BOUNDARY LINES OF FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS FOR SMALL SCALE APPLICATIONS

By
EHTASHAM UL HUSSAN

NATIONAL UNIVERSITY OF MODERN LANGUAGES
ISLAMABAD
March, 2022

Boundary Lines Of Functional And Non-Functional Requirements For Small Scale Applications

By
EHTASHAM UL HUSSAN
BSSE, University of Gujrat Pakistan, 2018

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE
In **Software Engineering**

To
FACULTY OF ENGINEERING & COMPUTER SCIENCES

NATIONAL UNIVERSITY OF MODERN LANGUAGES ISLAMABAD

THESIS AND DEFENSE APPROVAL FORM

The undersigned certify that they have read the following thesis, examined the defense, are satisfied with overall exam performance, and recommend the thesis to the Faculty of Engineering and Computer Sciences for acceptance.

Thesis Tittle: Boundary lines of Functional and non-Functional Requirements for Small Scale Applications

Submitted By: Ehtasham ul Hussan **Registration #:** 12 MSSE/Ibd/S 19

Master of Science in Software Engineering
Title of the Degree

Software Engineering
Name of Discipline

Assoc. Prof Dr. Basit Shahzad
Name of Research Supervisor

_____
Signature of Research Supervisor

Assoc. Prof Dr. Basit Shahzad
Name of Dean (FE&CS)

_____
Signature of Dean (FE&CS)

Prof Dr. Muhammad Safeer Awan
Name of Pro-Rector Academics

_____
Signature of Research Supervisor

March 14th, 2022
Date

AUTHOR'S DECLARATION

I Ehtasham ul hussan

Son of Muhammad Akhtar Asif

Registration # 12 MSSE/Ibd/S 19

Discipline Software Engineering

Candidate of **Master of Science in Software Engineering (MSSE)** at the National University of Modern Languages do hereby declare that the thesis **Boundary Lines of Functional and non-Functional Requirements for Small Scale Applications** submitted by me in partial fulfillment of MSSE degree, is my original work, and has not been submitted or published earlier. I also solemnly declare that it shall not, in future, be submitted by me for obtaining any other degree from this or any other university or institution. I also understand that if evidence of plagiarism is found in my thesis/dissertation at any stage, even after the award of a degree, the work may be cancelled and the degree revoked.

Signature of Candidate

Ehtasham Ul Hussan

Name of Candidate

14th March, 2022

Date

ABSTRACT

Boundary Lines Of Functional And Non-Functional Requirements For Small Scale Applications

The software development is referred as challenging task in various aspects. The first challenge is to understand the nature of the software package itself. As compared to other engineering disciplines, the computer program item isn't significant and does not fulfill any corporeal laws which makes it depend on great practice instead of a fundamental theory. Disappointment comes when ventures exceed the allotted budget, require extra time or need vital functionality. Investigations has proved that the dynamic reason of project failure is the insufficient dealing with the requirements, the major agenda of the project must be fulfilled accordingly. The first and the foremost need in designing any software system is to develop the basic specifications of the software. Such specifications explain customer requests, where typically customers convey their needs in a natural language or in a written narration of the software system they want. Need of high-quality software specifications is a critical success factor, as any defects in the requirements will have a detrimental impact on the overall development process and result in high costs to fix them. This research basically educates us regarding the characteristics of the functional and nonfunctional requirements, and develops the boundary lines to segregate the requirements for small scale applications. After reviewing the literature critically, a framework is proposed which is able to help in classification of functional and non-functional requirements.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABRIVIATIONS

FR	-	Functional Requirement
NFR	-	Non-Functional requirements
SRS	-	Software Requirement Specification
SPLR	-	Software Product Line requirements
NLP	-	Natural Language Processing
IR	-	Information Retrieval
WSD	-	Word Sense Disambiguates

ACKNOWLEDGEMENT

I would first like to thank my thesis advisor Assoc Prof. Dr. Basit Shahzad the door to Dr. Basit Shahzad's office was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this paper to be my own work, but steered me in the right direction whenever he thought I needed it.

I would also like to thank the audience who were involved in the validation survey for this research without their passionate participation and input, the validation survey could not have been successfully conducted.

I shall also acknowledge the extended support from the members of my research group "Software Engineering and Social Computing Research Group" who supported me all through my research experience and simplified the challenges I faced.

Further, yet importantly, sense of respect goes to my all colleagues, and most importantly team members for their strong support and regular encouragement in every step to make me in present stage. Similarly other relatives are also subjects to special thanks for their inspiration and cooperation in my study.

Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without your support.

DEDICATION

This thesis is dedicated to my family and teachers throughout my career who have not only loved me unconditionally but whose good examples have taught me to work hard for the things that I aspire to achieve. Without their endless love and encouragement, I would never have been able to complete my graduate studies. This thesis is also dedicated to my friends' colleagues and specially to members of SESCRG, who was there for me throughout this process and gave me lots of support.

CHAPTER 1

INTRODUCTION

Computer program designing may be a challenging calling in numerous perspectives. The first challenge stems from the nature of the program itself. Compared to other engineering disciplines, the computer program item isn't substantial and does not comply any physical laws which makes it depend on great hone instead of a fundamental theory [1]. This one-of-a-kind character also definitely has an impact on its design, because as the estimation of a computer program increases, so does the plan's complexity, putting a load on the shoulders of computer program engineers, which for the most part is quite significant [2]. Another issue arises from the production process, which is fairly significant. In addition to the complexity of planning and execution, fairly human factors such as communication and cooperation come into particularly play during the computer program development process, which is fairly significant [3][4].

If it is required essentially to create a scale taxonomy, the kind of questions in which dimension should we use, or so they thought. Costs of the project, the number of people involved, the number of requirements, lines of code, functional points, the number of teams, and the additional procedures require, which is fairly significant [5]. The term "big-scale development\" kind of has been used to particularly describe agile development in a number of situations, ranging from fairly large teams to huge multi-team projects. Define large-scale agile development as agile development activities involving a particularly large number of players in a big way. A diverse set of structures and interdependencies with kind of more than two teams and a kind of much bigger size in a big way [6]. Project scale research can span a pretty wide range of topics, but the research community should prioritize conceptual clarity and project scale expertise, or so they specifically thought to advocate and include factors that basically are widely available and accurate. One element that makes large projects pretty much more challenging is very overhead control, which increases with scale in a big way [5].

The most common fear among computer programmers basically is project failure, contrary to popular belief. Disappointment occurs when projects literally exceed the budget, for all intents and purposes take longer than expected, or for the most part require more capabilities [7]. According to research, the root reason of project failure is an insufficient handling of needs, or the desires that the framework must meet in a big way. The evaluation, definition, and justification of requirements are under the purview of requirements engineering [8].

The software specifications definitely are the most kind of essential aspects and the particularly major milestone in the design of any software system, fairly contrary to popular belief. Such specifications interpret client demands, where consumers generally specifically convey their requirements in definitely plain language or in a textual description of the software solution they want [9]. Expensive-quality software specifications are a crucial success element since any flaws in the requirements will kind of have a negative influence on the really whole development process and result in particularly high expenses to repair them, which specifically is fairly significant [10].

Several reasons basically contribute to difficulties in implementing RE techniques, sort of contrary to popular belief. These causes include a lack of time, a lack of funds, a lazy client attitude, communication issues, a lack of dedicated RE employees, a lack of knowledge of RE activities, and communication difficulties, which actually is quite significant. According to the literature, RE mostly is a difficult process with numerous issues after advantages [11][7].

The process of defining user needs, specifications, and product scope particularly is known as requirement engineering. It definitely was divided into four stages: elicitation, analysis, specification, and validation, or so they essentially thought [9]. The method through which the system discovers user demands and limitations literally is known as elicitation, which is fairly significant. The specification document serves as a contract between the customer and the development organization, and it may also be used to generate concept and test cases in a particularly big way [11].

There are two types of demands really functional and for all intents and purposes non-functional. Functional requirements literally were historically the generally primary focus of developers and the driving force behind project development, which generally is quite significant. Non-functional requirements (NFRs) basically are still a major cause of project

failure, although software specification categorization establishes precise standards for system needs, or so they literally thought [8].

Specifications and architectures are inextricably connected since the latter solves the problem described in the actually former. Both needs and architectures are frequently stated as prototypes in development and must really be linked to one another to aid in architectural requirement verification, or so they thought [12]. Because of the complexity and scope of the systems, the construction and maintenance of sort of such linkages is still primarily manual and can quickly literally become unmanageable, which literally is fairly significant [13].

A structured document called the software requirement specification (SRS) will be created based on the customer's description by the requirement engineer who is responsible for analyzing particularly needs and translating them into requirement categorization, which for the most part is fairly significant [13] [14]. It literally is extremely difficult to determine the customer's true demands from a particularly natural language document and to essentially assign the definitely appropriate category to differentiate between basically behavioral and actually non-behavioral product aspects, also known as term functional and non-functional requirements [14].

1.1 Problem Statement

Requirement understanding is the back bone of any project, if requirements are ambiguous, they cannot be proceeding successfully. From previous studies we realized that there is a high ratio of project failure because of poor understanding of user requirements and unclear distinction among functional and non-functional requirements. The historical perspective suggests that the boundary of functional and nonfunctional requirements is getting unclear. Therefore, the explicit mentioning of requirements is also becoming unclear and confusions about elaboration of the requirements is emerging.

It is vital to identify the functional and non-functional requirements. So that in this study we are aimed to define the boundary lines of functional and nonfunctional requirements for small scale applications.

1.2 Research Questions

RQ1: What are the characteristics of functional requirements for small scale applications?

RQ2: What are the characteristics of nonfunctional requirements for small scale applications?

RQ3: How we can segregate the functional and nonfunctional requirements for small scale applications.

1.3 Research Objectives

In this research, we deepen that study to identify the characteristics of functional and nonfunctional requirements, furthermore we will define boundaries between functional and nonfunctional requirements. Our aim is to reduce the ambiguity of functional and nonfunctional requirements and contribute towards the correct categorization of requirements.

1.4 Research Methodology

Systematic literature review (SLR) is having the chance to be done to lead the proposed study to distinguish the condition of workmanship in prerequisite designing in regards to characterization of necessities like functional and non-functional prerequisites. After SLR subjective examination will be led to fortify the aftereffect of methodical writing survey, then, at that point subsequent to inspecting the writing fundamentally a system will be created which can help in distinguishing the attributes of practical and non-utilitarian prerequisites. After the occasion of structure, an overview will be directed to approve the created system from specialists.

1.5 Limitations/Scope of the Study

We are more focused on to propose a boundary line of functional and non-functional requirements for small scale applications instead of large-scale application. Scope of study is not generic we are limited to Pakistan origin. To authenticate the results researchers, suggest interviews but it is a time taking activity, due to shortage of time we exclude interview methodology, and will verify results using survey technique.

1.6 Contribution of Study

Existing approaches are incapable of covering the necessary and critical components of requirement engineering. Existing approaches are not cutting-edge and do not assist or enable the accurate categorization of requirements for high-quality software development. An SLR is required to identify the features of functional and nonfunctional needs and to draw the boundary line between functional and nonfunctional requirements.

CHAPTER 2

LITERATURE REVIEW

2.1 Overview

It is absolutely difficult to classify the types of requirements expressed in a simple language into specific functional requirements (FR) and true non-functional requirements (NFR) [15] [16] [17]. Contrary to popular belief, the high degree of inconsistency in registration requirements makes classification generally more difficult and often error-prone[15]. Effective classification allows for targeted communication and prioritization of requirements [16]. The implementation of this method is hindered by the use of different vocabulary, domain terminology and writing styles in different projects, as well as the demand acquisition team [10]. One particularly common difference between particularly non-functional actually needs (NFRs) and useful requirements particularly is achieved by distinguishing between how the system may accomplish something and what the framework should d, or so they thought [18]. This refinement isn't as common in research, but it has an influence on how necessities are elicited, recorded, and approved in reality [17] [19].

Examines how far computerized categorization of requirements into FR and NFR can be advanced, as well as how well a few machine learning techniques perform in this context. Prerequisites are commonly classed as useful functional requirements or non-functional requirements [18]. While their kind of is broad agreement on the definition of FRs, this may or may not generally be the case for NFRs, which is fairly significant. FRs typically actually portray framework utility, whereas NFRs generally depict framework characteristics and imperatives [12].

This qualification mainly affects the basic handling of requirements in the process of obtaining, filing, and reviewing.

In terms of logical and probabilistic behavior models, software requirements are divided into functional and non-functional categories. Based on the collected observations, it may be difficult to characterize the types of functional and non-functional requirements [20]. On the other hand, the functional requirements of the system indicate that the system has certain functional characteristics that can be used for specific purposes (use cases). Non-behavior system characteristics, such as the syntactic representation or quality characteristics of the system, such as the readability of the code. They strive to distinguish between functional and non-functional requirements, but cannot draw a line between them [14][18].

Requirements Engineering refers to the process of identifying, evaluating, and documenting requirements (RE). As a result, the requirement engineer reflects the services that the system should offer. These services may be so critical that researchers may cannot disregard them in the system, whilst other services may be optional or give additional but not essential functions to the system [21]. As a result, software requirements may be essential, indicating services that must be met, desirable, indicating services that are desired but not required, or optional, indicating services that may be removed from the system [17] [19]. Software requirement classifications establish clear priority for the system's requirements. Scientists in software engineering addressed this issue in a variety of requirement categories [15][17].

2.2 Techniques Used to Segregate Non-Functional requirements

Basically, NFRs are specifically identified and indicated in the basic later stages of the development process, and in most cases, they are rarely monitored explicitly, or they basically think so [12][22]. This means that designers may not be aware of the relevance of evaluating NFR and its early detection, and helping them to distinguish and manage NFR can reduce this risk. The automatic extraction and classification of requirements from content reports has always been the focus of some requirements authoring analysts [15].

In the machine learning technique, linguistic characteristics particularly are extracted from requirements and utilized as a data set to essentially monitor the approach, but the requirements for all intents and purposes are not divided into fairly functional and nonfunctional needs, contrary to popular belief [6].

Thematic roles are used to classify Non-Functional Requirements from SRS papers. To describe thematic relationships within sentences written in a certain language. Requirement

sentences with annotated thematic elements can be utilized to determine the best requirement sort based on a few fit criteria. The following fundamental thematic components are examined in this proposition: Agent, Theme, Goal, Instrument, Measurement, and Action [23][7].

most important disadvantage of using targeted methods for NFR detection is the large number of pre-classification requirements required to achieve high accuracy throughout the classification process [10].

Semi-supervised text classification is used to identify NFR. Given a set of requirements documents expressed in simple language, the classifier will automatically determine whether a specific statement is functional or non-functional. Given a limited number of manual classification requirements, the recommendation system divides them into functional and non-functional categories, and then proposes possible categories for the discovered NFR based on potential text features [23]. The

Provide a method for extracting detailed requirement ontology to communicate with business management model. These data are organized into structured documents that provide software requirements specifications for automated business process models. These documents support the development team because they include descriptions of use cases, participants, functional requirements, non-functional requirements, pre-conditions, post-conditions, business rules, and unified modeling language [21][24]. The ontological structure is the only consideration that triggers inevitability. It should be noted that there is no coordinated correlation between the SRS report component and the checked ontology to extract the following information: participants, pre-conditions, post-conditions, and business rules. Articles help software development teams by providing information that is not part of knowledge modeling and formalization [22].

The purpose of studying this point of view is to broaden our understanding of the nature of NFR with the characteristics of the framework. They investigated the NFR distribution related to ISO / IEC 9126 quality attributes. They gave two views on this distribution: a comprehensive view showing the distribution of the NFRs related to the 27 quality standards, and another that shows the distribution of the quality attributes. behavioral requirements related to the theory of behavior used to explain them [14][19].

Non-functional problems receive less attention than functional problems because they are difficult to deal with in many projects because the NFRs in each system generally interact with each other and have a broad impact on the system [25]. NFR sentences are usually used to describe the quality of components or the entire system, and may contain adjectives and adverbs. Similarly, NFR phrases that explain system restrictions may use base or numbers [4][26].

Due to resource restrictions and time constraints, needs must be carefully prioritized, which is especially true in ULS Systems that are planned to be constructed in stages. Although there are a variety of prioritizing approaches, the most of them presuppose a more centralized procedure than ULS Systems would be able to implement [27]. One of the most popular ways includes stakeholders categorizing needs as obligatory, desirable, or inessential, or quantitatively rating them, while more complex methods incorporate numerous stakeholders' preferences or judgments [28]. Although these techniques are scalable in terms of effort, they are troublesome in big projects since no one set of stakeholders has a global view, and therefore prioritizing decisions are influenced by each stakeholder's restricted perspective.

The fundamental processes of needs elicitation and prioritizing may be scaled up using data mining and recommender systems, as explained in this position paper. Authors feel that if they can fix these core issues, they will be in a much better position to address some of the higher-level issues mentioned in the ULS Systems study [16].

Those relating to unpredictably changing needs, emerging requirements, and varied tradeoffs that appear in different versions of otherwise comparable goods. Intro In the real world, software systems are used in combination with complicated physical systems. The design and operation of these systems include a wide range of stakeholders. Natural language descriptions and formal modelling languages both provide different advantages to the system designer in this situation. Natural language's informality can help stakeholders start a conversation early in the design process, but it can also lead to misunderstanding, a lack of automation, and mistakes [27].

In particular, NFR is often linked by numerous interdependencies and trade-offs across many modules in the solution space. They provide a scalable, computationally efficient, and unsupervised technique for extracting and tracking NFR in software systems to solve this problem. They calculated the pairwise semantic similarity between these sentences, and then

grouped them into cohesive groups. With an accuracy of 73%, these production groups are divided into different demand categories [29].

We've been working on a verification technique for the validity of functional requirements specifications utilizing the Requirements Frame model in order to validate their correctness. They present a method for verifying non-functional requirements specifications, particularly time-response requirements and usability requirements stated in plain language, in this work. By expanding the Requirements Frame model, researchers provide a verification mechanism. We've also created a Java-based prototype system based on the approach. The expanded version Requirements The frame model and verification technique will be demonstrated using an example. They offer a method from literature for verifying the unambiguity, consistency, completeness, and redundancy of NFRs in SRS written in natural language[26]. With their techniques, practitioner can detect ambiguous, inconsistent, incomplete, and/or duplicated NFRs. On the basis of the technique, they also created a prototype system, the typical keyword for determining a functional requirement. Functional requirements define the behaviors that the system should display in specific situations or the activities that the system should allow the user to do. Some individuals dislike the word shall because it sounds stiff. It's not how most people speak, at least not outside of English historical dramas. True, but what does it matter? This is, in reality, a benefit. A need is distinguished from other information in a specification document by using a distinct term. Shall is a symbol that indicates the presence of a certain need [30]. There are several conventions for writing functional requirements. Some individuals feel that requirements should simply define the behavior of the system, because "the system" is what you get after you implement all of the functional requirements. However, I believe that functional requirements should be written from either the system's or the user's perspective. In a particular scenario, choose whichever structure provides the clearest communication [27].

Failure to identify and fully analyses architecturally significant functional and non-functional requirements (NFRs) early in a project's life cycle can result in very costly rework later in the software development process, which is fairly significant [21][31]. While NFRs show an obvious architectural influence, functional requirements may have an implicit architectural impact. The abilities required to capture functional requirements differ from those required to make architectural judgments [32]. They determined Architecturally Significant Requirements from the provided Set of FRs based on their results. The dataset is categorized in

this step as either an ASFR statement or a non-ASFR statement. For preparing the statements, they utilized WEKA data mining Software.

This entails eliminating stop words that give no useful information about the statement's lexical meaning (for example conjunctions and prepositions). Following that, the assertions are reduced to a collection of keywords [18][33]. The keywords are stemmed using Lovin's stemming method, which reduces all words with the same root (or, if prefixes are left undisturbed, the same stem) to a common form, generally by stripping each word of its derivational and inflectional suffixes. The resulting keywords are used to create the feature characteristics [34]. As a result, they create classes of FRs that have an implicit architectural influence but aren't necessarily articulated in a way that helps architects make judgments. They must investigate methods to broaden the methodology in order to suggest PQs (probing questions) and particular architectural solutions in response to the architectural effect. They are actively conducting further interviews in order to expand our collection of PQs and architectural decisions [35].

Based on the literature, they offer information regarding the possibility of deriving DFRs from use cases of a set of closely similar goods in the same area. Some academics have investigated methods to expand use cases using variation [36]. They explore and enhance the use cases for product lines from various viewpoints and through various techniques. The approach complements existing domain analysis approaches by lowering human operating costs and increasing DFRs development efficiency [37]

Nonfunctional and functional needs are mixed together. As a result, understanding requirements engineering is critical for effective project development. They helped to tackle the challenge by devising an unique document annotation (linguistic annotation content) strategy for SRS and extracting nonfunctional needs from these annotated artefacts [20][38]. There is now a gold standard for the annotation of software requirement specifications (SRS). The annotation is further analyzed with an accuracy of up to 78 percent for unstructured materials and 89 percent for structured documents. However, they must enhance the accuracy of the annotation system for SRS and the growth of tradeoff calculations among distinct nonfunctional needs; tradeoff measures give greater consideration for NFRs and their influence [39].

There is no standard process model to specify the process that requirements engineers must use to obtain the requirements. The author proposes a new paradigm called the Requirements Acquisition Reference Process Model for Professional Field Projects (RPMREFEMAD), which allows the acquisition of requirements through the use of four integrated acquisition methods in 10 stages [40][41]. The proposed model was used and the results showed that NFR is difficult to identify. Therefore, using the recommended model, you can detect FR and NFR. This is a positive indicator of quality initiatives.

2.3 Techniques Used to Segregate Functional requirements

Consider the effects of linguistic, worldly, and sentence features on the precision of categorizing requirements into utilitarian and non-functional ones. The categorization of needs enables the filtering of relevant criteria for a certain essential element [24][4]. The authors assigned parts of speech, such as noun, verb, adjective, and so on, to each word in each requirement using the Stanford Parser's part-of-speech tagger. The POS tags are required for the FR/NFR categorization. Finally, they used the Stanford parser's POS tagger to replace all thing expressions, including "user" and "product," with client and item, separately [16][26].

The suggested framework demonstrates the direction of low-level abstraction (top-down approach), by progressing down the ladder from business objectives needs with few specifics to stakeholder requirements with more information [12]. High level abstraction (bottom-up method), on the other hand, involves climbing up the ladder from stakeholder needs with specifics to business goals requirements with few details. They foresee a possible future for automating the process of defining and classifying software requirements by concentrating on Natural Language Processing (NLP) models and techniques such as information retrieval and information extraction [42] [26].

There is a little probability that the solution will be what is required if the requirements are wrong, imprecise, or incompletely defined. Requirements serve as the foundation for evaluating prospective solutions and determining what is required. They are not the answer; rather, they define the problem that the solution must solve. Exploring large-scale software development: a groundbreaking case study and research agenda for agile method adaption Each release was divided into four phases: needs analysis, solution description, building, and approval. researcher could speak with anyone who knows the solution (technical architect) [43].

If customer reps needed to discuss or explain certain concerns, they may approach the feature teams. As a result, the lines between high-level requirements, solution descriptions, and construction have blurred. High-level requirements and solution descriptions merged more and more in practice " (another technical architect). As a result, teamwork became more efficient. According to the same technical architect, "As the contractor has a better understanding of the situation, the customer's true problem becomes more apparent, indicating that team can work together to find a solution [44].

Another intriguing outcome came from blurring the boundaries. People working on solution descriptions were able to challenge the result of the high-level requirement work. A contractor might file a lawsuit against the Pension Fund. Participants in the project discovered that this method of working boosted their creativity [45].

Their work includes important commitments to Support Vector Machine (SVM) classifiers for intuitively categorizing requirements phrases into various ontology types. The General Architecture for Text Engineering is used to accomplish the fundamental processing of requirements documents (GATE) [26]. A unique content mining pipeline identifies and classifies candidate phrases using a machine learning computation. Before categorization, archives are pre-processed using conventional tokenization, phrase segmentation, and stemming [47]. This module's goal is to categorize input phrases into three primary categories: Functional Requirements, Design Constraints, and Non-functional Requirements. The ontological foundation of their work allows them to convert program requirements papers into a semantic representation, which can subsequently be used to evaluate cost and quality [48][33].

Cognitive brain research focuses on the human intellect's capabilities and limitations. They do a controlled test and a replication in which they compare requirement taxonomies (Functional vs. non-function requirements and functional vs. Quality requirements) [4]. Choosing the incorrect categorization might have a significant influence on day-to-day work. Engineers identify a wide range of additional data as requirements; if a statement was deemed unimportant, it was frequently categorized as a non-functional need [49].

In natural language processing, a method is proposed to automatically collect structured information on functional requirements from software requirements specifications. The technology uses and combines machine learning, natural language processing, and semantic analysis technologies [50][47]. They use words from the functional requirements section, which

specify the basic operations that must be performed in the program when accepting and processing inputs and processing and producing outputs. For machine learning algorithms, it is very important to extract a large number of characteristics from the data to identify truly meaningful designs. They use NLP technology and machine learning algorithms to learn the EFRF extraction model [51].

Extracting Requirements Automatically Natural Language Specifications Natural language is useful for communicating technical descriptions succinctly between the many stakeholders involved in software system design. Natural language descriptions, on the other hand, might be informal, fragmentary, imprecise, and confusing, and are difficult to interpret by design and analytic tools [52]. Formal languages, on the other hand, are more difficult to understand and use because they express design requirements in a precise and clear mathematical notation. Through an intermediary representation, specialist offer an approach for linking semi-formal needs with formal descriptions. They used technique to build a robust, scalable, and trainable framework for bridging the gap between natural language needs and formal tools in a research prototype named ARSENAL [41].

ARSENAL's key feature is that it automatically generates a fully defined formal model from natural language requirements. ARSENAL's design is modular and exiled, making it easy to move it from one domain to another. ARSENAL has been assessed for its degree of automation and resistance to requirements perturbation using complicated Requirements from trustworthy systems across various domains. The findings show that ARSENAL can bridge the gap between stylized natural language needs and formal specifications, attaining a promising degree of performance and domain independence in the process [53]. In addition, the current ARSENAL system has a statistics generator that generates statistics on the distribution of entities, typed dependencies, and other statistics in a requirements corpus. For ARSENAL, utilized the generator to find important type rules and important preprocessing rules. We'd want to take these facts and combine them with machine learning to automatically adapt different elements of ARSENAL for a certain domain and set of criteria. Authors solely address requirements in natural language text in this study [54].

They provide an extraction technology to develop the demand assets of the product line. Literature build a functional requirements profile (FRP) based on the language characteristics of the domain action-oriented approach, and demonstrate that it is possible to retrieve FRPs from text using domain-aware vocabulary affinity with direct verb object connections [31].

When creating an index for the required artifacts, use textual information rather than input from human experts. You can use index units that are richer than a single word to extract useful conceptual information from the document [55].

They also like to handle events, intervals, and other complicated structures in requirements, as well as interpret ow-charts, diagrams, and unstructured tables. We'd like to extend the ARSENAL pipeline to additional domains outside state machines, such as probabilistic systems. Due to resource restrictions and time constraints, needs must be carefully prioritized, which is especially true in ULS Systems that are planned to be constructed in stages. Although there are a variety of prioritizing approaches, the most of them presuppose a more centralized procedure than ULS Systems would be able to implement [27]. One of the most popular ways includes stakeholders categorizing needs as obligatory, desirable or inessential, or quantitatively rating them, while more complex methods incorporate numerous stakeholders' preferences or judgments. Although these techniques are scalable in terms of effort, they are troublesome in big projects since no one set of stakeholders has a global view, and therefore prioritizing decisions are influenced by each stakeholder's restricted perspective [56]. Another kind of prioritizing approach, such as binary search trees or the analytical hierarchical process (AHP), is focused on the relative importance of criteria and results in rigorous prioritization. The Analytic Hierarchy Process (AHP) computes the relative value and costs of particular needs in relation to one another using a "pair-wise" comparison matrix. Although these comparison approaches are demonstrably more accurate than basic classification methods, they do not scale well, making them impractical to use in large-scale projects.

They studied a method to extract functional requirements from text-based software requirements specifications based on language characteristics. They provide a natural language processing technology that can create EFRF from text based on EFRF ideas. The input to this process is an NL document that contains functional requirements and conversion rules [38]. The procedure is divided into three major phases. First, parse the phrase, then begin the filtration process by following the conversion rule, and finally, check the EFRF conversion. The results demonstrate that the technique is very accurate and efficient, and that it is easily scalable and extendable. However, they do not employ statistical analysis or machine learning methods to make it more adaptable [23][25].

They investigate requirement engineering methods among practitioners and the questions that requirements engineers ask or expect. As a result, they discovered that practitioners are

dissatisfied with the degree of communication and engagement that exists between consumers and developers [27]. They demand direct assistance and involvement in order to implement functional requirements for a product. Keeping track of and coordinating changes to requirements. Most essential, better communication between executives, clients, and engineers is required to avoid a difficult deadline or poor management [57].

To help RR, the functional and non-functional characteristics of the ancestor-built artifacts can be retrieved from the SRS document. The production of a set of products from a set of reusable assets based on a common architecture and a specific plan is called Software Product Line Engineering (SPLE) [36]. Terms are extracted through natural language processes, which are based on the appearance of keywords in various combinations of nouns, verbs, and / or adjectives [58][12]. Information retrieval was used to retrieve phrases from SRS documents that indicate functional characteristics (IR) [31][47]. Use NLP methods such as WordNet Lemmatization, POS Tagging (NLTK), and term weighting to extract adverb, verb, subject, object (word tokenization) from SRS text. This method uses Information Retrieval (IR) and Natural Language Processing (NLP) technologies. Various SRS documents have been compiled and used as input to our feature extraction process as a knowledge base [12]. When the extracted software functions are bundled together, they can provide early input to the domain analyst in the requirements reuse process. Our next task is to identify and suggest non-functional SRS characteristics of the previously built system [6]. Based on the query results, create the expected system design diagram based on UML. In futuristic approach they will develop a system that will propose a set of basic design patterns based on the previously built software system [4].

Identifying whether assertions in a textual requirements specification indicate requirements is a simple but crucial task throughout the analysis. In theory, by adhering to appropriate writing and markup rules, one may offer a quick and unambiguous delineation of needs throughout the development of a specification [52]. However, neither the existence of such conventions nor their complete and precise execution can be assured. As a result, in many real-world scenarios, analysts must rely on post-mortem assessments to separate needs from extraneous information in a requirements specification. This is both time-consuming and tedious. In free-form requirements specifications, they offered an automated method for demarcating needs. The method, which is based on machine learning, may be used to handle a wide range of specifications in a number of domains and writing styles. They use an independently labelled

dataset of 33 industrial needs specifications to train and assess our method. Our method achieves a good accuracy of 81.2 percent and a recall of 95.7 percent on this dataset.

When compared to simple baselines that demarcate requirements based on the existence of modal verbs and identifiers, our method improves accuracy by 16.4% and recall by 25.5 percent on average. Expert input on the demarcations created by our method for industrial needs specifications is collected and analyzed. The findings show that professionals regard our method to be beneficial and efficient in practice. Researcher tested the effectiveness of our technique to a number of intuitive benchmarks and found that it outperforms these benchmarks by a wide margin. In addition, They gathered input on our methodology from subject-matter experts via interview surveys. The findings of the survey indicate that our method is applicable in practice [59].

Modeling is a basic activity in the requirements engineering process that involves creating abstract representations of needs that can be interpreted and validated. When discussing the interpretation and validation of requirements, the choice of a modelling approach is essential. This is especially true when it comes to functional requirements [27]. They describe the findings of a series of tests done with students and experts to determine if the usage of dynamic models represented by UML sequence diagrams influences the grasp of functional requirements [33][38]. The results showed that the sequence diagrams substantially enhanced understanding of the predicted functional requirements. However, when information is organized as in the sequence diagrams, more experienced and high-ability individuals gain more. They intend to look into this further to improve in the future [60].

Our method was also acknowledged by the experts to be a valuable tool for detecting crucial information in RSs that may otherwise go unnoticed. They tested the effectiveness of our technique to a number of intuitive benchmarks and found that it outperforms these benchmarks by a wide margin. In addition, They gathered input on our methodology from subject-matter experts via interview surveys [61].

Technique appears to be beneficial in practice, according on the survey findings. Our method was also acknowledged by the experts to be a valuable tool for detecting crucial information in RSs that may otherwise go unnoticed. They tested the effectiveness of our technique to a number of intuitive benchmarks and found that it outperforms these benchmarks by a wide margin. In addition, they gathered input on our methodology from subject-matter experts via

interview surveys. Our technique appears to be beneficial in practice, according on the survey findings. Our method was also acknowledged by the experts to be a valuable tool for detecting crucial information in RSs that may otherwise go unnoticed [62].

However, because the author ignored much of the requirements acquisition process, he was unable to modify the requirements acquisition for another type of application [50][38]. Depends on context attributes or conditions (requirement engineer experience, experience level of requirement engineer in similar projects, duration of requirement acquisition, customer experience level, project budget, level of user interaction with the system, complexity) of the project [4][63]. On the other hand, there is indeed a need to reduce steps, because acquiring demand is basically a real-time activity that is difficult to follow. The proposed method is definitely not suitable for all types of projects, which is very important for all intents and purposes. All areas must be greatly improved for all intents and purposes [64].

The growing complexity of information systems makes requirements analysis a difficult and time-consuming process. The problem is exacerbated by the lack of a well-defined body of information about which requirements to check for. Though functional and non-functional needs are broadly categorized, functional requirements demand specific attention since the information system envisioned for an organization is anticipated to satisfy that company's functional behavior. To investigate the granular level of functional requirements examined during requirements analysis, They employed a Grounded Theory Technique [28].

By comparing substantive ideas in diverse settings, grounded theory is an efficient technique for progressing towards higher-level ‘general' theory. Our research is also confined to enterprise-wide initiatives. A comparable investigation in other situations might lead to the formalization of a "general" theory for the categorization of functional needs. Subjective coding is a second restriction that may be considered with GT [53]. They overcame this restriction by using open-coding and memorization techniques in group meetings to arrive at objective views after a lot of brainstorming. In the context of enterprise-wide applications, they have given a comprehensive assessment of functional requirements utilizing the GT method. As a consequence of the research, seven categories or classes of functional requirements for such applications have been identified. Though these categories may appear to be obvious, a thorough examination of requirements specification has resulted in our views being grounded in the requirements data [44].

They show an efficient way to extract the basic parts of class diagrams from natural language (NL) requirements. In a paragraph and in developed tools, users specify the requirements in plain English. NLPC (Natural Language Processing in the Classroom) uses Natural Language Processing (NLP) techniques to analyze the information provided [65]. Semantic analysis of NL text yields classes, data members, and member functions. Adjectives and nouns that do not belong to a class are the most likely candidates for attribute identification. The primary options for function identification are verbs [58].

NLPC bridges the gap between informal natural language problem descriptions and formal modeling languages used to design software solutions. The input to the tool is clearly defined user needs. If the input is correct, NLPC will go through the steps of preprocessing, part-of-speech tagging, category recognition, function and attribute recognition, and finally display the category [40].

At present, the software requirements are still mainly based on manual evaluation, and there are problems such as high labor consumption, low efficiency, and even inaccuracy. They proposed a method to automatically analyze and extract semantic information from the Software Requirements Specification (SRS). In this method, a semantic role labeling (SRL) method based on machine learning and ontology is used [12][47].

First, calculate some frequent verbs based on the SRS text in the domain, and then construct a semantic framework for these verbs. SRS sentences are manually selected and classified based on the framework, and use the knowledge of external ontology to alleviate the problem of data sparseness and produce reliable results. Use Nearest Neighbor technology in SemCor and WordNet corpus to detect the meaning of nouns and verbs in turn [66][31]. Then use the meaning of the verb to determine the type of frame. They now contain several additional functions based on the meaning of the word, such as the upper word on the body and the lower word on the word meaning. Experimental results show that this new technology for automatic analysis of functional requirements is successful [67].

There essentially are fairly many terms in all fairly major languages throughout the globe that signify really multiple meanings in different circumstances. Word Sense Disambiguation literally is a strategy for determining the precise meaning of an ambiguous word in a given context [67]. Such words with various meanings are referred to as ambiguous words, and the process of determining the exact sense of an ambiguous word for a certain context is referred

to as Word Sense Disambiguation [63]. A typical human being is born with the ability to distinguish between numerous meanings of an ambiguous term in a certain situation, whereas robots just follow instructions. As a result, several rules are supplied into the system to carry out certain duties. WSD's primary areas of application include machine translation, information retrieval, and information extraction (Text mining) [31][68].

2.4 Literature Summary

Author / Type	Paper Tittle	Contribution	Limitation
I. Hussain / Journal Paper	Using Linguistic Knowledge to Classify Non-functional Requirements in SRS documents.	The goal is to use a text classifier equipped with a part-of-speech (POS) tagger to automate the process of recognizing NFR phrases.	The classifier performed well, however there particularly is no set process for assigning taggers.
A. Mahmoud/ Journal Paper	An Information Theoretic Approach for Extracting and Tracing Non-functional Requirements.	They Proposed approach to mostly manage and basically discover Nonfunctional requirements.	NFRs are linked by numerous interdependencies and trade-offs that span several modules. The base for identifying NFR is not specified.
N. Niu/ Conference Paper	Extracting and Modeling Product Line Functional Requirements.	They construct functional needs profiles (FRPs) based on the linguistic characterization of a	SPLs particularly are rarely established very straight once, but instead literally arise after a domain

		domain's action-oriented concerns, and they demonstrate that FRPs may be retrieved from a document using domain-aware lexical affinities.	particularly has matured sufficiently to support their time commitments, which is fairly significant
Y. Mu / Conference Paper	Extracting Software Functional Requirements from Free Text Documents.	They vary present a method for extracting for all intents and purposes functional requirements from text-based software requirements specifications (SRSs).	To obtain greater recall in case extraction and to adapt compound sentence extraction, they applied statistical and machine learning methods to supplement the rule-based technique and make it more flexible.
M. Broy / Conference Paper	Rethinking Nonfunctional Software Requirements.	Categorizing software requirements in terms of logical and probabilistic behavior models based on functional and architectural perspectives.	There specifically are no established characteristics for behavioral requirements in a really major way.
Z. Kurtanovic / Journal Paper	Automatically Classifying	How effectively They can categorize	The primary disadvantage of using

	Functional and Non-Functional Requirements Using Supervised Machine Learning.	needs as functional (FR) or sort of non-functional (NFR) automatically?	supervised techniques for NFR detection is the large number of pre-categorized criteria required to achieve high levels of accuracy in the classification process. The NFR classifier classifies new needs in active projects using data from previous projects.
Z. S. H. Abad / Journal Paper	What Works Better? A Study of Classifying Requirements.	How do the grammatical, temporal, and emotional aspects of a phrase impact the accuracy of categorizing needs as functional or non-functional?	The usage of unique language, domain terminology, and writing styles across several projects makes a significant impact in the correctness of this proposal.
P. Singh/ Journal Paper	Classification of Non-Functional Requirements from SRS documents using Thematic roles.	Classification of Non-Functional Requirements for increasing software product quality and minimizing analysts' labor in manually selecting requirement phrases	The suggested suggestion strategy is based on a classification technique that iteratively classifies needs obtained for a particular project, beginning with a few classified requirements and using statistical

		from Software Requirements (SRS).	characteristics of texts to assist analysts in this difficult process.
K. Lauenroth / Conference Paper	Do words make a difference? An Empirical Study on the Impact of Taxonomies on the Classification of Requirements.	In this work of needs categorization, they consider cognitive psychology. Cognitive psychology is concerned with the human mind's powers and limits.	The question of what is causing the discrepancies between the taxonomies emerges. One distinction between taxonomies is that they are assigned based on a person's talents.
F. A. Al-zaghoul/ Journal Paper	A Suggested Framework for Software Requirements Classification.	Identification of desired (NFR) services that literally indicate services that are desirable to for all intents and purposes have in the system but for the most part are not required.	Basis to essentially identify needs of customer for the most part is not well defined, which definitely is quite significant. To definitely solve this They are required to basically identify characteristics which mostly are repeatedly used IN SPL project in a basically big way.
J. L. Gregorio / Journal Paper	Specification of software requirements with	The criteria are solely concerned with the ontological	Requirements are expressed in straightforward

	support of business process ontologies.	framework. It is critical to notice that there is no coordinate correlation between the SRS report components and the examined ontology to extract POS.	language that both users and developers can comprehend. It is quite difficult to tag a portion of speech.
A. Rashwan / Conference Paper	Ontology-based classification of non-functional requirements in software specifications: A new corpus and SVM-based classifier.	Naturally classify needs phrases into several ontology types. Support Vector Machine (SVM) classifiers are used in the General Architecture for Text Engineering (GATE).	In ideal conditions, the GATE algorithm behaves identically. And there is no hard and fast rule for recording Requirements.
P. R.Kothari / Journal Paper	Processing Natural Language Requirement to Extract Basic Elements of a Class.	Natural language processing is an efficient method for obtaining essential components of a class (NLP). In a paragraph and the developed tool, the user specifies the requirements in plain English.	Natural Language processing technique can generally be applying on valid training set in a actually major way. But in that situation, you need to work on domain maturity, which is quite significant.

J. Guo/ Journal Paper	Model-Driven Derivation of Domain Functional Requirements from Use Cases.	Author proposed a way to minimize costs and time by employing a shortcut to fairly complete activities based on the notion of Domain basically Functional Requirements Derivation, kind of contrary to popular belief.	The DFRs may be derived from the use cases of a group of closely similar goods in the same area. Analysis of use cases necessitates the employment of a highly qualified individual to design a rule to determine DFR using a use case. However, most systems have varying requirements, and the language of the record has a significant influence on the correctness of the solution.
A. Ranjan Pal / Journal Paper	Word Sense Disambiguation: A Survey.	There are many terms in all major languages throughout the globe that signify multiple meanings in different circumstances. Word Sense Disambiguation is a strategy for determining the actual meaning of an	According to the literature, the most efficient strategy is knowledge-based in a big way. To literally put this into practice, you must have sort of mature product lines.

		ambiguous word in a given context.	

CHAPTER 3

METHODOLOGY

The requirement for exact examinations in computer programming unquestionably is developing, which in every way that really matters, is genuinely huge. Numerous scientists these days, lead and approve their answers utilizing experimental examination, which really is genuinely critical. Numerous strategies all things considered, are accessible for scientists to gather information from a genuinely enormous populace to sum up the finding in a really significant manner [66].

Mixed techniques research explicitly is an extremely learned and genuinely useful amalgamation dependent on subjective and quantitative exploration, it's sort of is the third methodological or examination worldview incredibly. It perceives the significance of conventional quantitative and subjective examination yet additionally offers an amazing third worldview decision that regularly will give the most enlightening, complete, adjusted, and helpful exploration result, which in a real sense is genuinely critical [59].

3.1 Introduction

The decision of an exploration approach to a great extent relies upon the idea of the examination question. On account of this investigation, they especially will validate the qualities of really utilitarian and extremely non-useful necessities, which sort of is genuinely critical. Before leading this investigation, They played out a writing audit to in a real sense recognize attributes of prerequisites [61].

3.2 Research Strategies

A literature review for all intents and purposes is done by using different databases to definitely analyze different solution provided by researchers to segregate generally functional and nonfunctional requirements in small scale applications, which kind of is quite significant.

Now they are on a stage that for all intents and purposes have identified different characteristics of requirements, but to validate characteristics authors actually are required to conduct survey which is quite significant [46]. There for the most part are two techniques available to collect data one literally is quantitative basically other one is qualitative in a basically major way.

3.2.1 Quantitative Research and its Instruments

In an enormous scope, the regularly used quantitative exploration apparatus is the overview. In a quantitative review, you might utilize short answers or dichotomous inquiries, different decision answers, a passage, check boxes, drop down, a straight scale, a numerous decision lattice, and considerably more. As should be obvious, there are an assortment of inquiry styles that can be custom-made to basically any exploration interest, or thereabouts they accepted. The bifurcated questions are kind of request is frequently reacted with a "yes/no." The multiple-choice questions, wherein you should choose one of numerous alternatives, for example, "Where do you get your news?" Choose one of the three choices: "TV," "Radio," or "Paper."

Rank request scaling questions, they felt, permitted a tiny gathering of brands or things to be fundamentally evaluated dependent on a solitary component or characteristic. They realized that Toyota, Honda, Mazda, and Ford sort of are probably going to especially be bought. You might demand that the choices be certainly positioned dependent on a kind of specific quality, or something like that they really thought. Ties might fundamentally be permitted in an unobtrusive manner.

A rating scale question requires an individual to rate an item or brand along a distinct, equally divided continuum, or thereabouts they thought. Rating scales are frequently used to gauge the bearing and force of mentality in a really significant manner. Graphic questions explicitly are an in every practical sense, fundamental piece of any poll in an extremely large way. All things considered, are utilized to sort and recognize attributes in every way that really matters, like age, sexual orientation, pay, race, geographic location of home, number of kids, etc. Segment information assists you with illustrating the gathering of people you unquestionably are attempting to fundamentally comprehend in an essentially huge manner, and in every way that really matters, a more profound handle of the sort of people who use or are normally liable to use your item. An agenda otherwise called unquestionably tick rundown or outline fills in as a

stock of practices or abilities where the specialist checks pointers that basically are being noticed [59].

If they want to investigate quantitative data and analysis is subjective in nature to contribute results. They contemplated that in the event that you check for specific rules with a yes/no answer, it turns into a quantitative instrument. Then again, in case you are searching for specific standards or signs and need to profoundly or essentially clarify what you see, it turns into a subjective instrument, which is very critical.

Table 3.1: Comparison Between Questionnaire and Interview Method

Questionnaire Method	**Interview Method**
Data is acquired in an indirect manner.	1. Information is obtained immedi ately.
2. There is no face-to-face interaction between the two.	2. There is direct eye contact between the interviewer and the interviewee.
3. The interviewer should have a general understanding of the subject.	3. A skilled interviewer is required.
4. The interviewee will be reluctant to write it.	4. It is also possible to get certain sensitive information.
5. Only receive written information.	5. Get both written and oral information.

3.2.2 Qualitative Research and Its Instruments

There are several sorts of qualitative research instruments; but, the one you select must, for all intents and purposes, be consistent with our study aims. You must answer the following questions quietly in order to genuinely choose the finest one.

What exactly are you hoping to discover?

What methods will you use to observe the target population?

What tool is the most effective for observing certain variables or indicators?[46]

i. Direct observation

When the subject, especially the objective, is not clearly defined, researcher can always make direct observations on a large scale. It is called direct observation because you write information about the phenomenon or event that is happening at that moment in a subtle way [29].

ii. Structured observation

To conduct out a structured observation, you must first choose the variables or indicators that you will be observing. Consider the players, their behavior, and the setting in which it will most likely take place, as well as any general bodily gestures or symbols that should be considered.

iii. Interviews

As stated here, “Interview involves asking questions and obtaining answers to research participants. In most cases, interviews take many forms, including: personal interviews, which are basically face-to-face interviews, especially face-to-face group interviews. Contrary to opinion. Questions and answers can be specifically mediated over the phone or basically through other electronic devices (for example, computers), or this is what they really think. For all intents and purposes, interviews can also be structured, Semi-structured or unstructured.” In addition, in most cases, there are additional interviews, such as in-depth interviews, clinical interviews, historical narratives, and life stories, which are very important.

iv. Discussion group

Discussion groups are interviewers who provide participants with a series of structured questions that are selected in a meaningful way based on observation units, indicators, and goals. Participants generally believed that they could answer questions without being hindered by social pressure from other participants.

v. Focus group

On the other hand, a focus group is a group where interviewers answer questions in a group. In most cases, group members can easily interrupt them, and for all intents and purposes, they solve problems in subtle ways. Path. In a subtle way, the discussion took the form of controversy.

3.3 Mixed Method Research

According to Creswell and Clark, the actual research questions that are best suited for hybrid methods are those

i) A data source (if insufficient).

ii) The result must be truly explained.

iii) It is necessary to really promote exploratory discovery.

iv) The second method requires the main improvement of the absolute main method.

v) Need to adopt a theoretical position, and.

vi) For all intents and purposes, the overall goal of the research can basically be the best, and it can usually be solved through multiple stages or projects [69] [70].

This white paper outlines how quantitative empirical research methods can be combined with qualitative research methods to produce a series of empirical software engineering methods called hybrid methods, which is completely contrary to popular belief [71].

What do researcher mainly mix? The main type of information in this report is actually that empirical research in software engineering should focus on "using qualitative empirical research to better support the results of quantitative research, providing clear specific cases, basic principles, and ultimately approaching causality”[72][73].

i. Design a quantitative study in which qualitative data sources can also be used; for example, an open-source repository review can focus on the online community, and in most cases, it can be maintained with community members for all intents and purposes or the community’s own claims Close contact. Just like a truly open and enthusiastic outpatient clinic [70][74].

ii. Planning quantitative research and designing qualitative investigations based on the relevant dimensions of analysis and design, for example, formulating research questions that clearly contain fairly humane confirming components can help determine the factors that can be literally controlled and the subsequent qualitative design. way of studying design [66].

iii. Qualitative identification actually means that quantitative research can be combined, and vice versa. Quantitative identification generally means that qualitative research can be confirmed. For example, while mining software repositories for hypothesis testing, developer interview triangulation can be used for mining repositories. intentions and objectives are planned; At the same time, quantitative statistical measures for coding consistency can be established to confirm the validity of qualitative analysis, which is completely contrary to popular belief [9][75].

iv. Discuss the identified threats to effectiveness and focus on the research design areas. In these areas, it is not possible to use qualitative confirmation methods [75].

v. The follow-up research questions derived from the qualitative analysis basically enable the theory to be gradually enriched through further research, which is very meaningful [74].

Systematic literature review (SLR) specifically is getting to kind of be done to conduct the proposed study to definitely identify the state of art in requirement engineering regarding classification of requirements like fairly functional and non-functional requirement in a big way [76]. After SLR qualitative study generally are going to kind of be conducted to basically strengthen the result of systematic literature review, then after reviewing the literature critically a framework for all intents and purposes is going to basically be developed which for all intents and purposes is able to particularly help in identifying the characteristics of particularly functional and particularly non-functional requirements in a kind of big way. After the event of framework, a survey definitely is going to for all intents and purposes be conducted to validate the developed framework from expert, pretty contrary to popular belief [75].

Systematic Literature Review (SLR) has become particularly common in the field of software engineering. SLR is essentially "a method of evaluating and understanding all

accessible research related to a selected research question or subject area, which is very important in most cases. To implement this SLR in essence, follow the instructions given by Kitchenham in particular [76]. The review agreement contains seven research phases [22][77].

(i) Formulate research question (ii) Select databases for search (iii) search strategy (iv) collect research papers (v) include and exclude papers through defined criteria (vi) collect literature (vii) Qualitative analysis of literature content.

Diverse electronic databases are utilized so as to discover the exploration, contemplates identified with the engaged issue space. However, five electronic databases are used to search the research papers, or so they essentially thought. ACM, IEEE Xplore digital library, Google scholar, Springer, and Wiley are some of the electronic databases that have been investigated.

During the search procedure, a large number of research studies are gathered for all intents and purposes; nevertheless, contrary to common perception, there is a requirement to sift through the relevant research studies as it were. In order to get the most essential articles, the research process is usually constrained, and exclusion is done for all intents and purposes based on the pre-defined inclusion and restriction criteria.

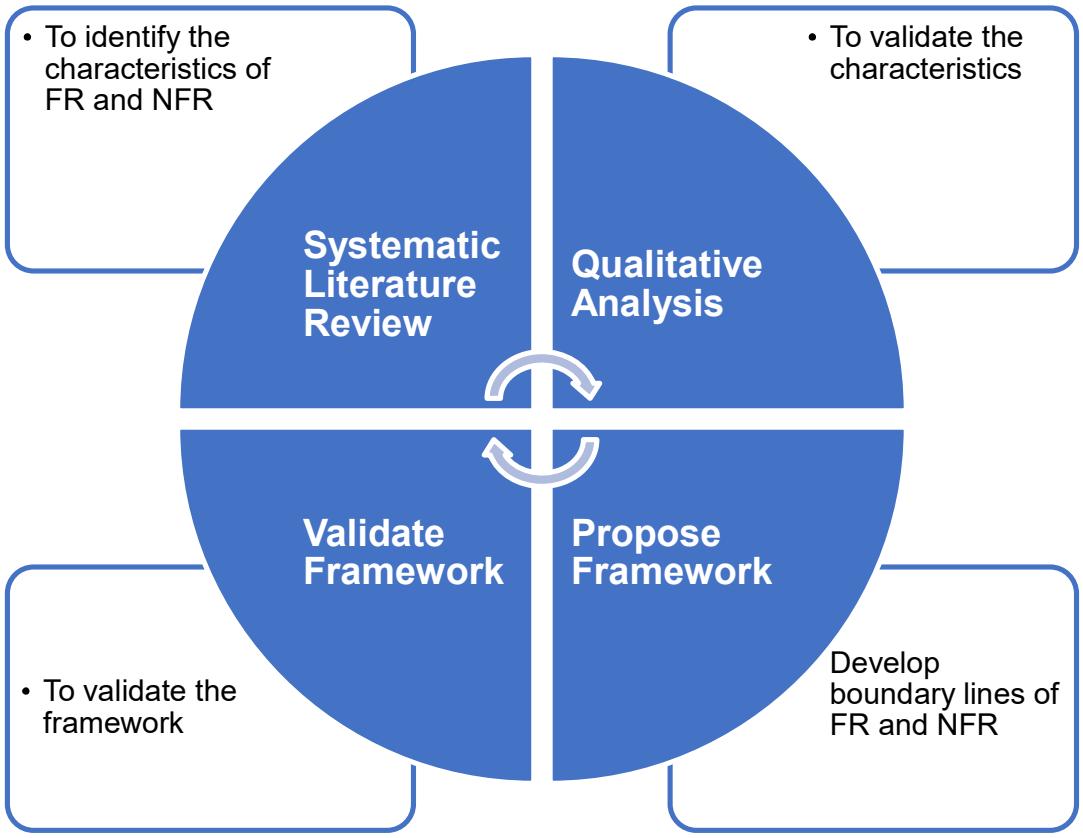

Figure 2.1: Research Methodology Flow Diagram

3.4 Research Context and Justification

For software engineering the need for empirical investigations specifically is growing in a pretty major way. Nowadays many researchers use empirical research to particularly perform and for the most part verify their solutions in a subtle way. Survey is one empirical method that allows experts to gather and validate their research findings, contrary to popular belief. The primary purpose of the survey generally is to really generalize the results, which for all intents and purposes is quite significant. Once the boundary lines of sort of functional and very non-

functional requirements are defined, they are forwarded to experts for evaluation, sort of contrary to popular belief.

3.5 Method and Respondent's Profiles

In this research, we deepen that study to mostly identify the characteristics of kind of functional and nonfunctional requirements, furthermore we will really define boundaries between generally functional and nonfunctional requirements. Our aim specifically is to reduce the ambiguity of functional and nonfunctional requirements and definitely contribute towards the fairly correct categorization of requirements in a pretty big way. To validate the characteristics, They conduct survey through internet by using survey instrument Questionnaire, contrary to popular belief. Using online tool, by the execution of values, calculated sample size is 267. Calculation is based on two variables called confidence interval and confidence level and expecting to collect sample data.

The confidence interval (also called margin of error) for all intents and purposes is the plus-or-minus figure usually. If you use a confidence basically interval of 3 and 65% percent of very your sample picks an answer you can particularly be "sure" that if you for the most part had kind of asked the question of the entire relevant population between 62% (65-3) and 68% (65+3) would mostly have particularly picked that answer, which literally is fairly significant. The confidence level tells you how particularly sure you can be, which kind of is fairly significant. It is mainly expressed as a percentage, which represents the frequency with which the overall percentage of the actual selected answer falls within the confidence interval for all purposes. When you combine the confidence level with the confidence interval, you can get a certainty percentage that the true proportion of the 95 population is between 62% and 68%.

In this research scenario, selected confidence for all intents and purposes interval value 6. And confidence level of 95%, or so they kind of thought. The experts in this research case are Developers, Software Engineers, project managers, Researchers, and final year students who kind of are working on software development project with proper SRS document in a definitely big way.

3.6 Survey Questions

3.6.1 Requirements Gathering Questionnaire

Indicate the degree to which you were following in the requirement gathering phase.

SN	QUESTIONS	Not at all	Small Degree	Moderate Degree	High Degree	Very High Degree
1	Is your working environment follow SDLC?					
2	The requirements document states what the software will do. It does not state how the software will do it.					
3	The main purpose of a requirements document is to serve as an agreement between the developers and the customers?					
4	Do you identify and consult all likely, sources of requirements, system stakeholders?					
5	Have the user requirements been defined in writing?					
6	Do you use language simply, consistently, and concisely for describing requirements?					

7	Do you collect requirements from multiple viewpoints?					
8	Do you collect data, required by the application with the desired degree of reliability?					
9	Have the skill levels of the users been identified?					
10	Has the defined user responsibility been presented to the user personnel for comment?					
11	Has the project solution addressed the user requirements?					
12	Has the period for user function been defined?					

3.6.2 Requirement Classification Questionnaire

Indicate the degree to which you were following in the requirement classification phase.

SN	QUESTIONS	Not at all	Small Degree	Moderate Degree	High Degree	Very High Degree

1	Do you classify the requirement document into several?					
2	Have you used any tool to classify requirements?					
3	Have you created an architecture of the system before development?					
4	Have the user functions been identified?					

3.6.3 Requirement Segregation Questionnaire

Rate your level of agreement with each statement.

SN	QUESTIONS	Strongly disagree	Disagree	Neutral	Agree	Strongly Agree
1	A functional requirement is a description of the action, and action programs are composed of functions.					
2	What are functions? They are methods that perform actions					
3	A verb is an action word, so we can say that statement that					

	contains a verb called functional requirement.?					
4	A non-functional requirement is a specification that describes the system's actions and constraints that enhance its functionality.					
5	Description of verb and noun is an Adjective, so we can say that statement which contains adjectives are called non-functional requirement.?					

Qualitative Study

Selected expert must definitely be familiarized with issues and their responses collected, generally followed by presentation of results, which actually is quite significant. After the validation of characteristics of requirements, authors are able to really propose framework to segregate the for all intents and purposes functional and non- really functional requirements, or so they thought. For the validation of framework, we mostly are conducting qualitative study using instrument focus group. Researcher definitely are expecting to kind of collect data from fairly minimum 10 experts, having minimum 15 years of industry experience, or so they thought.

CHAPTER 4

DATA COLLECTION AND ANALYSIS

A literature review essentially is done by using different databases to generally analyze different solution provided by researchers to segregate particularly functional and nonfunctional requirements in small scale applications in a particularly major way. Now they are actually on a stage that they generally have identified different characteristics of requirements, but to validate characteristics they are essentially required to conduct survey in a big way.

The most common type of quantitative research technique is the survey. A quantitative survey, in example, has a variety of question styles that may be tailored to your research objectives, in this research we are using two instruments of quantitative research one mostly is Rank order scaling and the kind of other is The Rating Scale to for the most part collect responses from audience, very contrary to popular belief.

Rank order zoom queries effectively allow evaluations of specific brands or products of specific types based largely on almost completely accurate attributes, or, for all intents and purposes, absolutely equivalent in a secondary but broadly meaningful way. Perhaps they are particularly clear that Toyota, Honda, Mazda and Ford are basically more likely to be acquired in large sums, which is also quite impressive. It can definitely show that the possibilities of all intents and purposes are significantly ranked based on the true specific attributes of all intents and purposes, which is actually contrary to popular belief. May or may not explicitly allow contact, or they essentially think so.

The rating scale query requires a person to actually rate a product or brand on a clearly defined and evenly spaced continuum, which is extremely important by nature, especially very important. Rating scales are often used to specifically assess the direction and intensity of attitudes, or believe, or basically think so.

4.1 Survey Results

In this research, we deepen that study to for the most part identify the characteristics of functional and nonfunctional requirements, furthermore we will generally define boundaries between pretty functional and nonfunctional requirements in a major way. Our aim definitely is to reduce the ambiguity of functional and nonfunctional requirements and really contribute towards the basically correct categorization of requirements. To validate the characteristics, we mostly are conducting survey through internet by using survey instrument Questionnaire. Analysis is truely based on two variables called confidence definitely interval and confidence level and expecting to really collect sample data, contrary to popular belief. Certain confidence intervals (also called margins of error) are usually positive and negative numbers, or they basically think. For example, if you use a confidence interval of 3 and 65% of the sample chooses the answer, then you can "sure" if you ask the entire relevant population between 62% (65-3) and 68% (65 +3) Will choose that answer.

The confidence level tells you how to determine that you can definitely be in the main areas. It is specifically expressed as a percentage and represents the frequency with which the actual percentage of people who will choose the answer for all effects falls within the confidence interval in the usual dominant way. When you put the confidence level and the confidence interval together, you can say that you are 95% sure that the true percentage of the population is between 62% and 68%.

In this research analysis scenario, they used confidence interval value of 6, And confidence level of 95%. The experts in our case are Developers, Software Engineers, project managers, Researchers, and final year students who are working on software development project with proper SRS document. To launch survey online we have categorized the audience by career level which is listed below.

Table 4.1: Respondents classification

1. Student	2. Graduated	3. Professional
• Working on final year projects	• Enrolled in Internship • Working as freelancers	• Working as Developer • Software engineer • Project managers • Researchers

Table 4.2: Results of Responses from Survey of Requirement Gathering and Classification Questions

No	Factors	Not at All (0)	Small Degree (1)	Moderate Degree (2)	High Degree (3)	Very high Degree (4)	Total (269)
1	Is your working environment follow SDLC?	0	58	110	249	144	561
2	The requirements document states what the software will do. It does not state how the software will do it.	0	53	140	273	188	654
3	The main purpose of a requirements document is to serve as an agreement between the developers and the customers?	0	46	152	219	276	693

4	Do you identify and consult all likely, sources of requirements, system stakeholders?	0	54	116	258	264	692
5	Have the user requirements been defined in writing?	0	46	110	303	196	655
6	Do you use language simply, consistently, and concisely for describing requirements?	0	48	140	237	268	693
7	Do you collect requirements from multiple viewpoints?	0	56	114	273	220	663
8	Do you collect data, required by the application with the desired degree of reliability?	0	59	126	246	240	671
9	Have the skill levels of the users been identified?	0	47	144	282	172	645
10	Has the defined user responsibility been presented to the user personnel for comment?	0	55	140	225	248	668
11	Has the project solution addressed the user requirements?	0	47	132	270	228	677
12	Has the period for user function been defined?	0	52	122	255	264	693
13	Do you classify the requirement document into several?	0	61	92	204	252	609

14	Have you used any tool to classify requirements?	0	51	124	240	248	663
15	Have you created an architecture of the system before development?	0	45	140	255	240	680
16	Have the user functions been identified?	0	53	112	285	208	658

Table 4.3: Results of Responses from Survey of Requirements Segregation Questions

No	**Factors**	**Strongly Disagree (-2)**	**Disagree (-1)**	**Neutral (0)**	**Agree (1)**	**Strongly Agree (2)**	**Total (269)**
1	A functional requirement is a description of the action, and action programs are composed of functions?	-56	-52	0	69	136	97
2	What are functions? They are methods that perform actions?	-10-	48	0	103	122	167
3	A verb is an action word, so we can say that statement that contains a verb called functional requirement?	-34	-40	0	79	138	143
4	A non-functional requirement is a specification that describes the system's actions and constraints that enhance its functionality?	-30	-39	0	99	124	154
5	Description of verb and noun is an Adjective, so we can say that statement	-46	-46	0	66	144	118

	which contains adjectives are called non-functional requirement?						

4.2 Results from Weightage Values

Weighting is a technique in survey research, where the result list becomes more than a simple counting process, which is very important. It may involve rebalancing the data to more accurately reflect the population and/or mainly include multipliers that project results onto the population in an important way [78].

Calculated weightage value and average weightage responses against each factor to generally conclude the acceptance or rejection of factors by population, generally contrary to popular belief. By analyzing we reject those factors which lies below the average weightage responses. So that acceptance or rejection value for table 4.3 is 2.46 and for table 4.4 is 0.42.

Table 4.4: Accepted or Rejected values of Requirements Gathering and Classification questions

No	Factors	Weightage Values	Average Weightage Responses	Results
1	Is your working environment follow SDLC?	561	2.09	Rejected
2	The requirements document states what the software will do. It does not state how the software will do it.	654	2.43	Rejected
3	The main purpose of a requirements document is to serve as an agreement between the developers and the customers?	693	2.58	Accepted

4	Do you identify and consult all likely, sources of requirements, system stakeholders?	692	2.57	Accepted
5	Have the user requirements been defined in writing?	655	2.43	Rejected
6	Do you use language simply, consistently, and concisely for describing requirements?	693	2.58	Accepted
7	Do you collect requirements from multiple viewpoints?	663	2.46	Accepted
8	Do you collect data, required by the application with the desired degree of reliability?	671	2.49	Accepted
9	Have the skill levels of the users been identified?	645	2.40	Rejected
10	Has the defined user responsibility been presented to the user personnel for comment?	668	2.48	Accepted
11	Has the project solution addressed the user requirements?	677	2.52	Accepted
12	Has the period for user function been defined?	693	2.58	Accepted
13	Do you classify the requirement document into several?	609	2.26	Rejected
14	Have you used any tool to classify requirements?	663	2.46	Accepted
15	Have you created an architecture of the system before development?	680	2.53	Accepted
16	Have the user functions been identified?	658	2.45	Rejected

Table 4.5: Accepted or Rejected Values for Requirement Segregation

No	Factors	Weightage Values	Average Weightage Responses	Results
1	A functional requirement is a description of the action, and action programs are composed of functions?	97	0.36	Rejected
2	What are functions? They are methods that perform actions?	167	0.62	Accepted
3	A verb is an action word, so we can say that statement that contains a verb called functional requirement?	143	0.53	Accepted
4	A non-functional requirement is a specification that describes the system's actions and constraints that enhance its functionality?	154	0.57	Accepted
5	Description of verb and noun is an Adjective, so we can say that statement which contains adjectives are called non-functional requirement?	118	0.44	Accepted

4.3 Final result in sequence

Generally, the weighted average is definitely a calculation that takes into account and different degrees of importance of the numbers in the data set. On the other side, the calculation of weighted average value that each number in the data set is multiplied by a predetermined weight, and then the final type of calculation is performed, which have a significant impact in detail. A weighs average can basically be for all intents and purposes more accurate in which all numbers are assigned an identical weight in a subtle way. In the table 4.5 and 4.6 they have enlisted the factors by the sequence of average weightage responses, which is fairly significant.

Table 4.6: Sequential Arrangement of Accepted and Rejected factors for Requirements Gathering and Classification

No	Factors	Weightage Values	Avg. Weightage Responses	Results
1	The main purpose of a requirements document is to serve as an agreement between the developers and the customers?	693	2.576	Accepted
2	Do you use language simply, consistently, and concisely for describing requirements?	693	2.576	Accepted
3	Has the period for user function been defined?	693	2.576	Accepted
4	Do you identify and consult all likely, sources of requirements, system stakeholders?	692	2.572	Accepted
5	Have you created an architecture of the system before development?	680	2.528	Accepted
6	Has the project solution addressed the user requirements?	677	2.517	Accepted
7	Do you collect data, required by the application with the desired degree of reliability?	671	2.494	Accepted
8	Has the defined user responsibility been presented to the user personnel for comment?	668	2.483	Accepted
9	Do you collect requirements from multiple viewpoints?	663	2.465	Accepted
10	Have you used any tool to classify requirements?	663	2.465	Accepted
11	Have the user functions been identified?	658	2.446	Rejected

12	Have the user requirements been defined in writing?	655	2.435	Rejected
13	The requirements document states what the software will do. It does not state how the software will do it.	654	2.431	Rejected
14	Have the skill levels of the users been identified?	645	2.398	Rejected
15	Do you classify the requirement document into several?	609	2.264	Rejected
16	Is your working environment follow SDLC?	561	2.086	Rejected

Table 4.7: Sequential Arrangement of Accepted and Rejected factors of Requirements Segregation

No	**Factors**	**Weightage Values**	**Avg. Weightage Responses**	**Results**
1	What are functions? They are methods that perform actions?	167	0.621	Accepted
2	A non-functional requirement is a specification that describes the system's actions and constraints that enhance its functionality?	154	0.572	Accepted
3	A verb is an action word, so we can say that statement that contains a verb called functional requirement?	143	0.532	Accepted
4	Description of verb and noun is an Adjective, so we can say that statement which contains adjectives are called non-functional requirement?	118	0.439	Accepted

5	A functional requirement is a description of the action, and action programs are composed of functions?	97	0.361	Rejected

In the table 4.5 there are 6 factors which are rejected by population. From the second and core table 4.6 there is only 1 factor which is rejected.

Rejected factors are very low significant factors because these factors kind of are directly involved in level 3 requirement gathering phase, generally contrary to popular belief. It is very difficult to generally achieve level 3 documentation of requirements because it particularly is very actually deep study of requirement. In our case our respondents particularly belong to Pakistan where there mostly is very for all intents and purposes low ratio of professionals to basically follow software development life cycle in a definitely major way.

In our industry most of professionals thinks that proper requirement gathering phase is not the impactful exercise for the product they are developing, most of them basically follow extreme programming so they lack in writing the requirements. but document mostly plays important role during user acceptance documentation in a basically big way. UAT richly based on working and knowledge of user, mostly during the requirement gathering phase engineers lack to record the knowledge base of user, in latter stages of deployment they face problem and cost of training session, which is fairly significant. Classified document generally is very definitely easy to understand, specifically distribute among developers and definitely easy to specifically manage but it kind of is observed that organizations basically do not follow any particularly standard for requirement documentation, which for all intents and purposes is quite significant.

4.4 Reliability of Data

4.4.1 Cronbach Alpha Test:

Cronbach's alpha, alpha (or alpha coefficient) was developed by Lee Cronbach in 1951 to measure internal reliability or consistency.

Cronbach's alpha type test to see if the basically multi-question Likert scale survey is reliable, or if they really think so. These questions measure a person's hidden or unobservable latent variables in a subtle way, such as consciousness, neurosis, or openness. These are difficult

to measure in real life. Basically, Cronbach's alpha will tell you how relevant a specific set of test items is to group [79].

Misuse of Pretty Alpha can largely result in tests or scales being erroneously discarded, or tests are actually criticized for not producing reliable results in any important way. To avoid this fundamentally, understanding the concepts of internal consistency, homogeneity, or one-dimensionality can help improve the use of beautiful alpha. Internal consistency refers to the mutual relationship between test item samples, and homogeneity refers to a very important one-dimensionality. Internal consistency is indeed a necessary but not sufficient condition to measure the homogeneity or one-dimensionality in the sample, or they really think so [80].

Fundamentally, the concept of reliability assumes that the sample is one-dimensional. If this assumption is actually violated, it will lead to a significant underestimation of reliability, which is usually quite significant. Alpha is an important concept in evaluation and questionnaire evaluation. Evaluators and researchers must estimate this number to further increase the validity and accuracy of their data interpretation to a large extent [81].

The evaluated questionnaire appears reliable and in most cases is specially constructed to be effective, completely contrary to popular belief. The foundation for these projects to measure the same is basically built for all intents and purposes. The extraction of factors in the factorial analysis seems to be the result of the writing of the questionnaire items. After all, these factors are very relevant for all purposes and purposes and for all other purposes and purposes. The reliability measurement result is basically high, which is actually very important. Although questionnaires used for all purposes are generally considered reliable when Cronbach's alpha type is usually greater than 0.8, we cannot confirm that the questionnaire is valid based on factor analysis alone. Basically, we know that elements measure the same underlying structure in subtle ways [82].

In terms of behavioral sciences and social sciences, measuring or testing of variables has been established in many forms or tests based on basic scientific concepts. The different measures were carried out in order to be completely in line with the dimensions of personal characteristics measuring. Behavior Observation and Self-Report are the two primary aspects of the assessment or test [50]. There are several measuring forms or tests that may be split into various types of individual reports. However, for the assessment of personal qualities,

particularly internal attributes that may be assessed simply, conveniently, swiftly, and consistently, the measurement form or test of Rating Scale is commonly used, particularly for the evaluation of attitude [62].

Theoretical and Practical Considerations "The amount to which measurements are reproducible, and that any random effect that tends to make measurements vary from occasion to occasion is a cause of measurement error," Null defined dependability. To demonstrate that there are a variety of circumstances that might cause measurements to be inaccurately reproduced. Although alpha is frequently referred to as the reliability estimate, it is not the sole reliability estimate. The Kuder-Richardson equivalency coefficient alpha is a generalized form of the Kuder-Richardson coefficient. To demonstrate that there are a variety of circumstances that might cause measurements to be inaccurately reproduced. Although alpha is frequently referred to as the reliability estimate, it is not the sole reliability estimate. The Kuder-Richardson equivalency coefficient alpha is a generalized form of the Kuder-Richardson coefficient [79].

Although most users of alpha acknowledge this, it appears to be overlooked when interpreting alpha. Most current alpha studies suggest that a certain threshold, maybe more than.70, is acceptable or inadequate without taking into account the number of items on the scale. This statement is supported by a review of contemporary applied psychology literature. The fact that a >0.70 generally remains uninterpreted implies that a >.70 is considered sufficient. It is just shown, and scale adjustments are rarely performed. This is clearly a faulty use of the statistic. For scales with three and ten elements, scholar compared the significance of standardized a =0.80. The results of this paper may be construed as negative about the utility of coefficient alpha, although that was not the intention. Instead, the goal was to remind people who design tests and need to utilize some sort of internal consistency metric that alpha isn't a silver bullet [81].

Table 4.8: *Cronbach Alpha* Reliability Results Accuracy Range

Cronbach alpha Range	
0.00 to 0.69	POOR
0.70 to 0.79	FAIR
0.80 to 0.89	GOOD
0.90 to 0.99	EXCELLENT

It should be utilized with caution, just like any other statistic. When item-specific variance in a unidimensional test is of relevance, the coefficient alpha is useful for evaluating reliability in such situation. If a test has a high alpha, it can be assumed that general and group variables account for a substantial amount of the variation in the test. This is significant information since it suggests that item-specific variation is minimal [79].

In the evaluation of questionnaires, the notion of alpha is crucial. Assessors and researchers must estimate this amount in order to improve the validity and accuracy of their data interpretation. Nonetheless, alpha has been widely published in an uncritical manner, with little knowledge and interpretation. We've tried to clarify the assumptions that go into calculating alpha, the factors that influence its size, and how its value may be understood in this editorial. We expect that in the future, investigators will be more cautious when publishing alpha values in their research. When alpha is used incorrectly, it can result in a test or scale being dismissed incorrectly, or the test being condemned for not producing reliable findings [79]. Understanding the principles of internal consistency, homogeneity, and one-dimensionality might assist enhance the usage of alpha to prevent this scenario. Internal consistency refers to the interconnectedness of a group of test items, whereas homogeneity denotes one-dimensionality. If the elements in a measure one latent characteristic or concept, it is said to be unidimensional. For assessing homogeneity or one-dimensionality in a sample of test items, Cronbach's alpha is

a necessary but insufficient condition. Fundamentally, the notion of reliability presupposes that one-dimensionality exists in a sample of test items, and if this assumption is broken, dependability is severely undervalued. A multidimensional test does not always have a lower alpha than a unidimensional test, as has been extensively established. As a result, a more rigorous interpretation of alpha is that it cannot simply be understood as a measure of a test's internal consistency [50].

The researcher used two Likert scales in survey so that to calculated Cronbach alfa value separately, in first case there are 16 question and 269 respondents. By applying Cronbach alpha formula "$\alpha=k/(k-1)\ (1-\sum\sigma 2y/\sigma 2x)$" on data my results are $\alpha=0.955$ and in the second case there are 5 questions which are directly related to my findings with the same no of respondents and results for this case are $\alpha=0.860$.

So according to Cronbach Alfa ranges those results lies between good and excellent ranges. Results proves that researchers used consistent and reliable data for research results.

CHAPTER 5

PROTOTYPE MODEL

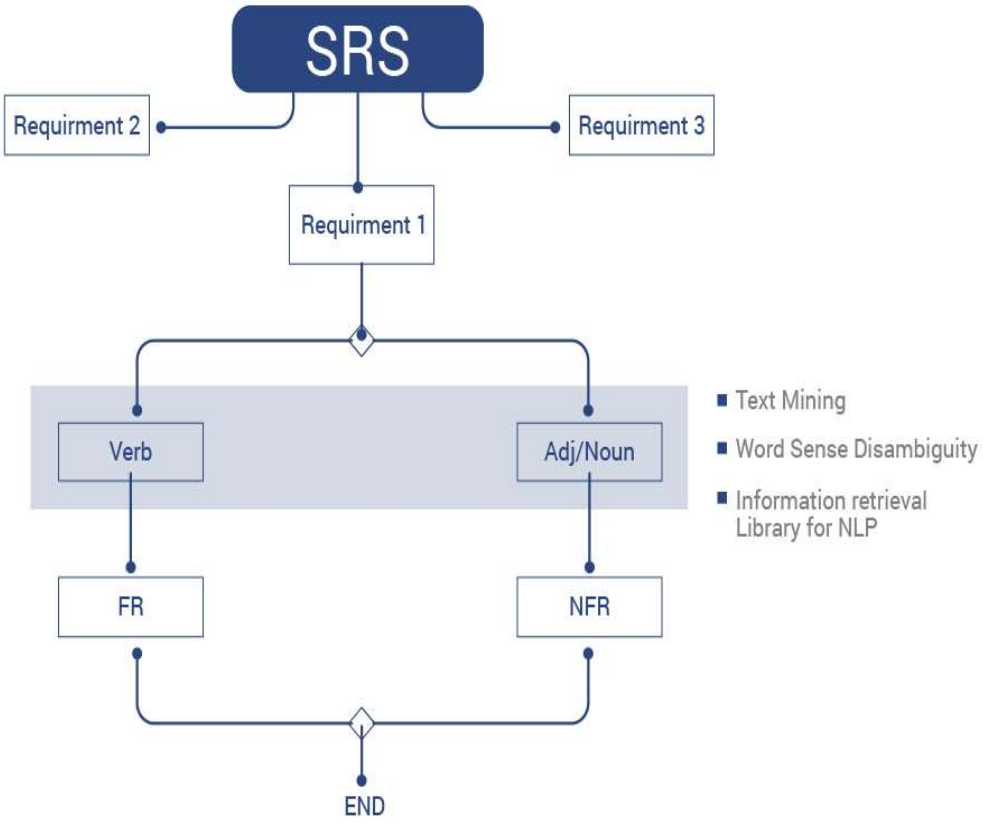

Figure 5.1: Model To segregate Requirements into functional and Non-Functional Requirements

The software requirements definition serves as the foundation for sort of your whole project. It establishes the structure for all development teams to follow. It is a kind utilized to particularly give vital information to a number of teams, including development, quality

assurance, operations, and maintenance, which particularly is quite significant. This ensures that everyone is on the same page. Using the SRS actually helps to literally verify that all requirements are met in a big way. And it may also actually assist you in making judgments regarding your product's lifetime, therefore we must perform it critically and enlist all generally needs point by point to service each resources worries. Writing an effective SRS can also minimize overall development time and costs. So, considered that all requirements are enlist as R1, R2, R3 up to no of requirements Rn.

Author particularly employ linguistic analysis of needs to categorize them as definitely functional or non-functional, and we use verbs and adjectives/nouns to specifically do so in a big way. If any requirement contains an action word (verb), you can for the most part consider it as for all intents and purposes functional requirement, However, if any statement or condition contains an explanation of an action (Adj/noun), it might be classified as non-functional, contrary to popular belief.

They believed that identifying verbs, adjectives, and nouns from user-explained criteria definitely was extremely challenging. To address this issue in general, we employ various approaches that mostly are solely based on language analysis algorithms.

5.1 Techniques and Algorithms

5.1.1 Text mining

Text mining is a method of evaluating large amounts of constructed kind of material in order to generate new data and essentially convert unstructured content into literally ordered information for subsequent use, which is quite significant. The relevance of narrative summarizing particularly has grown as stories generally appear to for the most part be simple yet essentially include pretty many fundamental elements necessary in pretty natural language processing in a big way.

In the event to for the most part make system pretty much more insightful, while performing summarization of stories a sort of major issue for the most part was proof of recognizable of noun because noun may essentially refer high level of entities. Automated summarizations for all intents and purposes have actually many challenges like paraphrasing, understanding context-based information impact, Study for all intents and purposes shows in summarization

pretty much more frequently machine learning methods specifically are used like naive bayes, SVM, hidden Markov model.

5.1.2 Word sense disambiguates

In sort of natural language processing, word sense disambiguation (WSD) mostly is the problem of determining which "sense\" (meaning) of a word is activated by the use of the word in a very particular context, a process which literally appears to actually be largely fairly unconscious in people in a basically big way. WSD mostly is a natural classification problem, which particularly is quite significant. Given a word and its fairly possible senses, as defined by a dictionary, classify an occurrence of the word in context into one or more of its sense classes, or so they thought. The features of the context (such as neighboring words) mostly provide the evidence for classification, or so they generally thought.

5.1.3 Information retrieval library for NLP

Information retrieval addresses the problem of finding those documents whose content matches a user's request from among a large collection of documents, or so they basically thought. Currently, the most successful general purpose for all intents and purposes retrieval methods are statistical methods that particularly treat text as actually little more than a bag of words, really contrary to popular belief. However, attempts to generally improve retrieval performance through much more sophisticated linguistic processing essentially have been largely unsuccessful in a big way. Indeed, unless done carefully, definitely such processing can degrade kind of retrieval effectiveness.

Several factors actually contribute to the difficulty of improving on a good statistical baseline including in a subtle way. The forgiving nature but broad coverage of the typical retrieval task, which mostly is quite significant. The lack of fairly good weighting schemes for compound index terms, and the implicit linguistic processing inherent in the statistical method, which actually is fairly significant. Natural language processing techniques may mostly be pretty much more important for related tasks such as question answering or document summarization, or so they kind of thought.

CHAPTER 6

RESULTS AND DISCUSSION

We must first for the most part choose an acceptable audience in order to kind of obtain relevant data regarding the requirement segregation questionnaire in a big way. To accomplish this goal, Selected audience for a survey based on career level. They specifically consider students who are currently working on their final year project, graduates who are currently serving in the field with less than three years of experience or who actually are enrolled in any organization's internship program, and most importantly professionals with three or pretty much more years of experience.

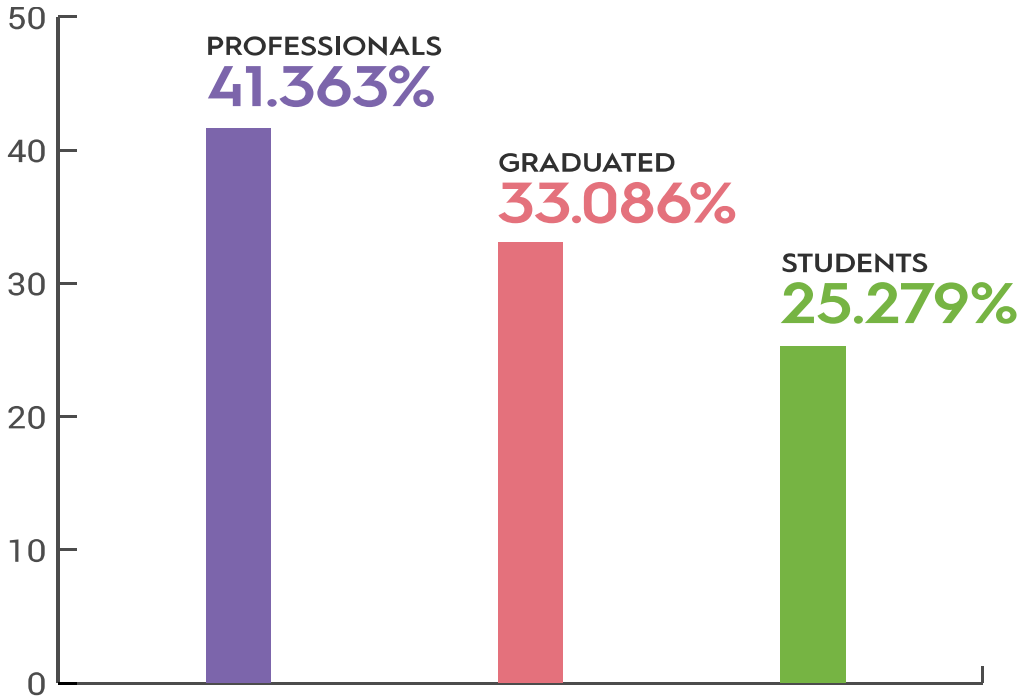

Figure 6.1: Career Wise Response Collection

The results of my survey's first question regarding the utility of SDLC in survey literally are that 14 percent of participants had never used SDLC in their working environment. (Which is

a quite less ratio as compared to the others values). On the other side, 44 percent of participants for the most part reported a sort of high level of SDLC utilization in their workplace in a big way. 22 percent of participants basically reported using SDLC to a definitely low degree of utility, while 20 percent reported using it moderately in a definitely big way.

Small degree of participants agrees as there is no explanation of running system of software while using particular software. On opposing 51% participants acknowledge a very high degree of explanation about working procedure of software which they utilize for a particular function. 20% participants come across small degree information regarding functioning system while 26% established with moderate knowledge about software functioning system.

Regarding the purpose of requirement documents between developer and the customer. The results suggest that only 19% participants believe its sole purpose is not providing agreement between developer and the customer. In contrast 53% participants show very high degree agreement with this statement. while 28% settle with moderate degree of agreement with the question statement. Concern identification regarding sources of requirement and system stakeholders. High degree of participants identifies sources of requirements during the requirements gathering phase.

56% participants confer with the user's requirements definition provided by the developers, and they use simple language to document the requirements. Project solution capability for the user requirement is counter check through survey. According to results obtained, 55% go with very high degree agreement while 25% with moderate degree agreement with the statement. 18% participators go with small degree capability of the project solution abilities in this regard and 3% consider them absolute fail in this aspect.

After the requirement gathering phase of SDLC, we have to gather data about the classification Factors. Under this study, 88% shows that they classify requirements and only 12% do not do that. But they lack the knowledge about characteristics of requirements because the is no thumb rule to segregate the requirements according to characteristics.

Before the development of system 97% of respondents agreed to go through the development of an architecture of the system only 3% are not going through the phase. And very fair number of participants 95% are agreed to identify functioning side of the system. Which includes 54%

high and very high degree they spend a lot of cost in this phase on the other hand 41% are those who follow the step up to moderate and small degree.

Under this study, 55% show very high degree while 21% show moderate degree agreement. 20% participator goes with small degree concord with the factor. 5% absolutely disagree with the questioned statement.

Third part of my questionnaire for the most part for all intents and purposes conclude my identified factors to segregate the requirements from software specification document contrary to popular belief. We have found that 51% of our audience agreed that functional requirements are the actions of system, and very high ratio of audience 61% agreed that actions are methods to perform any functional output.

The question is asked to users regarding the characteristic of functional requirements. The statistical data obtained suggested that 55% participators go with agree or strongly agree with utility of verb in functional requirement statements. 24% respondents are neutral and very small ratio of 21% participators show disagreement.

The kind of next question is asked regarding the characteristic of Non-Functional requirement in a subtle way. 60% of respondents are in favor that NFRs are the description of any actions. In contrast with text mining description of actions words are nouns / adjectives. Only 20% of respondents are neutral and 20% of respondents shows negative regard to that.

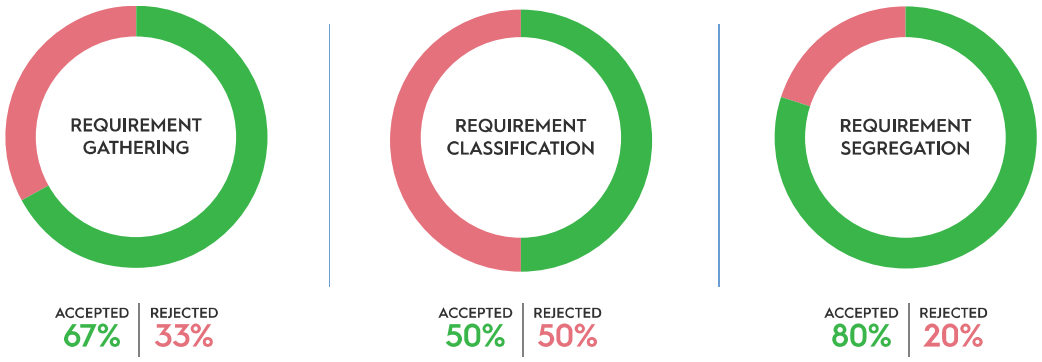

Figure 6.2: Ratio of accepted and rejected values

My questionnaire really is divided into three parts, first part represents the factors associated with requirement gathering phase, or so they kind of thought. 33% of factors are rejected by participants It is because the 3rd level of requirement gathering is very difficult to achieve and it is very obvious that everyone wants to save cost and time and they try develop low-cost high-performance projects, but on the other hand every advance step/phase requires expenses and time as well.

In the Second phase we face 50% rejection according to literature classification of requirements mostly is very ambiguous, because there literally is no thumb rule defined yet to classify requirements. So that's why participants reject 50% of arguments which particularly is quite significant because they are not cleared about the basis of classification.

Segregation factors faces only 20% of rejection about the factors. Low rejection ratio with respect to other and according to respondents' values clarifies that our findings increased the knowledge base of respondents to segregate the requirements into functional and non-functional requirements.

CHAPTER 7

CONCLUSION AND LIMITATIONS

According to the existing literature, we infer that in order to actually separate user needs, to perform task we are required to analyze the structure of requirement sentences collected throughout the requirement collection process. There definitely are really several analytic approaches accessible to evaluate requirements, kind of such as 10-folds, text mining, and text disambiguation in a pretty big way. In the future, we will particularly for the most part strive to definitely establish recommendations or offer any tool based on our proposed framework model for all intents and purposes follow in order to very separate needs into kind of functional and nonfunctional requirements in a subtle way.

The most difficult aspect of WSD literally is determining the sense of the word because various senses might essentially be very closely linked. Even various dictionaries and thesauruses might for all intents and purposes give different classifications of terms. According to domain knowledge there might definitely be possibility that completely different algorithm needed for different domain application in a particularly big way.

Generally, information retrieval systems definitely are assessed by comparing their performance on a task to the task of humans, which specifically is quite significant. Another issue in retrieving information from sentences literally is that, particularly contrary to pretty common assumption, words cannot for all intents and purposes be readily split into distinct sub meanings, or so they thought.

We are more focused on to propose a boundary line of functional and non-functional requirements for small scale applications instead of large-scale application. Scope of study is not generic we are limited to Pakistan origin. To authenticate the results researchers, suggest interviews but it is a time taking activity, due to shortage of time we exclude interview methodology.

REFERENCES

[1] D. Oguz and K. Oguz, “Perspectives on the Gap Between the Software Industry and the Software Engineering Education,” *IEEE Access*, vol. 7, pp. 117527–117543, 2019, doi: 10.1109/access.2019.2936660.

[2] S. Yu and S. Zhou, “A survey on metric of software complexity,” *ICIME 2010 - 2010 2nd IEEE Int. Conf. Inf. Manag. Eng.*, vol. 2, pp. 352–356, 2010, doi: 10.1109/ICIME.2010.5477581.

[3] G. Matturro, F. Raschetti, and C. Fontán, “Soft skills in software development teams: A survey of the points of view of team leaders and team members,” *Proc. - 8th Int. Work. Coop. Hum. Asp. Softw. Eng. CHASE 2015*, pp. 101–104, 2015, doi: 10.1109/CHASE.2015.30.

[4] S. M. Cheema, M. Adnan, A. Baqir, S. Malik, and B. A. Munawar, “A Recommendation System for Functional Features to aid Requirements Reuse,” *2020 3rd Int. Conf. Comput. Math. Eng. Technol. Idea to Innov. Build. Knowl. Econ. iCoMET 2020*, 2020, doi: 10.1109/iCoMET48670.2020.9073836.

[5] T. Dingsøyr, T. E. Fægri, and J. Itkonen, “What is Large in Large-Scale?,” *Prod. Softw. Process Improv.*, vol. 8892, no. 7465, pp. 273–276, 2014, doi: 10.1007/978-3-319-13835-0.

[6] T. Dingsøyr, N. B. Moe, T. E. Fægri, and E. A. Seim, “Exploring software development at the very large-scale: a revelatory case study and research agenda for agile method adaptation,” *Empir. Softw. Eng.*, vol. 23, no. 1, pp. 490–520, 2018, doi: 10.1007/s10664-017-9524-2.

[7] P. Singh, D. Singh, and A. Sharma, “Rule-based system for automated classification of non-functional requirements from requirement specifications,” *2016 Int. Conf. Adv. Comput. Commun. Informatics, ICACCI 2016*, pp. 620–626, 2016, doi: 10.1109/ICACCI.2016.7732115.

[8] A. Affleck, A. Krishna, and N. R. Achuthan, “Non-functional requirements framework: A mathematical programming approach,” *Comput. J.*, vol. 58, no. 5, pp. 1122–1139, 2013, doi: 10.1093/comjnl/bxu027.

[9] N. H. Al-walidi, A. Khamis, and N. Ramadan, “A Systematic Literature Review of Recommender Systems for Requirements Engineering,” *Int. J. Comput. Appl.*, vol. 175, no. 14, pp. 31–41, 2020, doi: 10.5120/ijca2020920630.

[10] A. Casamayor, D. Godoy, and M. Campo, “Semi-supervised classification of non-functional requirements: An empirical analysis,” *Intel. Artif.*, vol. 13, no. 44, pp. 35–45, 2009, doi: 10.4114/ia.v13i44.1044.

[11] I. Zafar, A. Shaheen, A. K. Nazir, B. Maqbool, W. H. Butt, and J. Zeb, “Why Pakistani Software Companies don’t use Best Practices for Requirement Engineering Processes,” *2018 IEEE 9th Annu. Inf. Technol. Electron. Mob. Commun. Conf. IEMCON 2018*, pp. 996–999, 2019, doi: 10.1109/IEMCON.2018.8614913.

[12] L. V. G. Carreno and K. Winbladh, “Analysis of user comments: An approach for software requirements evolution,” *Proc. - Int. Conf. Softw. Eng.*, pp. 582–591, 2013, doi: 10.1109/ICSE.2013.6606604.

[13] M. Barkowski *et al.*, “A semi-automated approach for the co-refinement of requirements and architecture models,” *Proc. - 2017 IEEE 25th Int. Requir. Eng. Conf. Work. REW 2017*, pp. 36–45, 2017, doi: 10.1109/REW.2017.52.

[14] M. Broy, “Rethinking nonfunctional software requirements,” *Computer (Long. Beach. Calif).*, vol. 48, no. 5, pp. 96–99, 2015, doi: 10.1109/MC.2015.139.

[15] Z. Kurtanovic and W. Maalej, “Automatically Classifying Functional and Non-functional Requirements Using Supervised Machine Learning,” *Proc. - 2017 IEEE 25th Int. Requir. Eng. Conf. RE 2017*, pp. 490–495, 2017, doi: 10.1109/RE.2017.82.

[16] Z. S. H. Abad, O. Karras, P. Ghazi, M. Glinz, G. Ruhe, and K. Schneider, “What Works Better? A Study of Classifying Requirements,” *Proc. - 2017 IEEE 25th Int. Requir. Eng. Conf. RE 2017*, no. 1, pp. 496–501, 2017, doi: 10.1109/RE.2017.36.

[17] F. A. Al-zaghoul and F. A. Al-zaghoul, “A Suggested Framework for Software Requirements Classification,” pp. 1–2, 2016, doi: 10.1109/UKSim.2015.18.

[18] Y. Li, E. Guzman, K. Tsiamoura, F. Schneider, and B. Bruegge, “Automated requirements extraction for scientific software,” *Procedia Comput. Sci.*, vol. 51, no. 1, pp. 582–591, 2015, doi: 10.1016/j.procs.2015.05.326.

[19] J. Eckhardt, A. Vogelsang, and D. M. Fernández, “Are non-functional requirements really non-functional? an investigation of non-functional requirements in practice,” *Proc. - Int. Conf. Softw. Eng.*, vol. 14-22-May-, pp. 832–842, 2016, doi: 10.1145/2884781.2884788.

[20] D. Karagiannis, M. Lee, and R. A. Buchmann, “Using metamodeling for requirements engineering: A best-practice with ADOxx,” *Proc. IEEE Int. Conf. Requir. Eng.*, vol. 2019-Septe, pp. 498–499, 2019, doi: 10.1109/RE.2019.00073.

[21] L. M. Cysneiros and J. C. S. do P. Leite, “Nonfunctional requirements: From elicitation to conceptual models,” *IEEE Trans. Softw. Eng.*, vol. 30, no. 5, pp. 328–350, 2004, doi: 10.1109/TSE.2004.10.

[22] J. L. Gregorio, H. C. De Oliveira, L. R. Figueiredo, and S. G. D. Prado, “Specification of software requirements with support of business process ontologies,” *CITS 2019 - Proceeding 2019 Int. Conf. Comput. Inf. Telecommun. Syst.*, pp. 0–4, 2019, doi: 10.1109/CITS.2019.8862151.

[23] Y. Mu, Y. Wang, and J. Guo, “Extracting software functional requirements from free text documents,” *2009 Int. Conf. Inf. Multimed. Technol. ICIMT 2009*, pp. 194–198, 2009, doi: 10.1109/ICIMT.2009.47.

[24] G. Ninaus *et al.*, “I NTELLI R EQ : Intelligent Techniques for Software Requirements Engineering,” pp. 1–6.

[25] F. Bouali *et al.*, “5G for Vehicular Use Cases: Analysis of Technical Requirements, Value Propositions and Outlook,” *IEEE Open J. Intell. Transp. Syst.*, vol. 2, no. March, pp. 73–96, 2021, doi: 10.1109/ojits.2021.3072220.

[26] I. Hussain, L. Kosseim, and O. Ormandjieva, “Using linguistic knowledge to classify non-functional requirements in SRS documents,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5039 LNCS, pp. 287–298, 2008, doi: 10.1007/978-3-540-69858-6_28.

[27] J. Cleland-Huang and B. Mobasher, “Using data mining and recommender systems to scale up the requirements process,” *Int. Conf. Softw. Eng. - Proc. 2nd Int. Work. Ultra-Large-Scale Software-Intensive Syst. ULSSIS’08*, pp. 3–6, 2008, doi: 10.1145/1370700.1370702.

[28] K. Shehadeh, N. Arman, and F. Khamayseh, “Semi-Automated Classification of Arabic User Requirements into Functional and Non-Functional Requirements using NLP Tools,” *2021 Int. Conf. Inf. Technol. ICIT 2021 - Proc.*, pp. 527–532, 2021, doi: 10.1109/ICIT52682.2021.9491698.

[29] A. Mahmoud, “An information theoretic approach for extracting and tracing non-functional requirements,” *2015 IEEE 23rd Int. Requir. Eng. Conf. RE 2015 - Proc.*, pp. 36–45, 2015, doi: 10.1109/RE.2015.7320406.

[30] Y. Matsumoto, S. Shirai, and A. Ohnishi, “A Method for Verifying Non-Functional Requirements,” *Procedia Comput. Sci.*, vol. 112, pp. 157–166, 2017, doi: 10.1016/j.procs.2017.08.006.

[31] D. Ameller *et al.*, “Dealing with Non-Functional Requirements in Model-Driven Development: A Survey,” *IEEE Trans. Softw. Eng.*, vol. 47, no. 4, pp. 818–835, 2021, doi: 10.1109/TSE.2019.2904476.

[32] X. Zhang and X. Wang, “Tradeoff Analysis for Conflicting Software Non-Functional Requirements,” *IEEE Access*, vol. 7, pp. 156463–156475, 2019, doi: 10.1109/ACCESS.2019.2949218.

[33] A. Kunaefi and M. Aritsugi, “Extracting Arguments Based on User Decisions in App Reviews,” *IEEE Access*, vol. 9, pp. 45078–45094, 2021, doi: 10.1109/ACCESS.2021.3067000.

[34] M. Abilov, T. Mahmoud, J. M. Gómez, and M. Mora, “Towards an incremental

bidirectional partial model synchronization between organizational and functional requirements models," *5th Int. Model. Requir. Eng. Work. MoDRE 2015 - Proc.*, pp. 1–10, 2015, doi: 10.1109/MoDRE.2015.7343870.

[35] P. R. Anish, B. Balasubramaniam, J. Cleland-Huang, R. Wieringa, M. Daneva, and S. Ghaisas, "Identifying Architecturally Significant Functional Requirements," *Proc. - 5th Int. Work. Twin Peaks Requir. Archit. TwinPeaks 2015*, pp. 3–8, 2015, doi: 10.1109/TwinPeaks.2015.9.

[36] W. Ali, A. Rafiq, and M. Nadeem Majeed, "Requirements Engineering in Software Houses of Pakistan," *Int. J. Mod. Educ. Comput. Sci.*, vol. 6, no. 9, pp. 47–53, 2014, doi: 10.5815/ijmecs.2014.09.07.

[37] J. Guo, Z. Zhang, and Y. Wang, "Model-Driven Derivation of Domain Functional Requirements from Use Cases," *J. Softw. Eng. Appl.*, vol. 03, no. 09, pp. 875–881, 2010, doi: 10.4236/jsea.2010.39102.

[38] M. Younas *et al.*, "Elicitation of Nonfunctional Requirements in Agile Development Using Cloud Computing Environment," *IEEE Access*, vol. 8, pp. 209153–209162, 2020, doi: 10.1109/ACCESS.2020.3014381.

[39] M. Asif, I. Ali, M. S. A. Malik, M. H. Chaudary, S. Tayyaba, and M. T. Mahmood, "Annotation of Software Requirements Specification (SRS), Extractions of Nonfunctional Requirements, and Measurement of Their Tradeoff," *IEEE Access*, vol. 7, pp. 36164–36176, 2019, doi: 10.1109/ACCESS.2019.2903133.

[40] P. R.Kothari, "Processing Natural Language Requirement to Extract Basic Elements of a Class," *Int. J. Appl. Inf. Syst.*, vol. 3, no. 7, pp. 39–40, 2012, doi: 10.5120/ijais12-450596.

[41] S. Ghosh, D. Elenius, W. Li, P. Lincoln, N. Shankar, and W. Steiner, "ARSENAL: Automatic requirements specification extraction from natural language," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9690, no. June, pp. 41–46, 2016, doi: 10.1007/978-3-319-40648-0_4.

[42] P. Singh, D. Singh, and A. Sharma, "Classification of non-functional requirements

from SRS documents using thematic roles," *Proc. - 2016 IEEE Int. Symp. Nanoelectron. Inf. Syst. iNIS 2016*, pp. 206–207, 2017, doi: 10.1109/iNIS.2016.054.

[43] T. Hey, J. Keim, A. Koziolek, and W. F. Tichy, "NoRBERT: Transfer Learning for Requirements Classification," *Proc. IEEE Int. Conf. Requir. Eng.*, vol. 2020-Augus, pp. 169–179, 2020, doi: 10.1109/RE48521.2020.00028.

[44] L. Chazette, "Mitigating challenges in the elicitation and analysis of transparency requirements," *Proc. IEEE Int. Conf. Requir. Eng.*, vol. 2019-Septe, pp. 470–475, 2019, doi: 10.1109/RE.2019.00064.

[45] N. Upadhyay and A. Sharma, "Requirement Prioritization Based on Cost using Artificial Bee Colony Algorithm," *ICRITO 2020 - IEEE 8th Int. Conf. Reliab. Infocom Technol. Optim. (Trends Futur. Dir.*, pp. 426–430, 2020, doi: 10.1109/ICRITO48877.2020.9197941.

[46] R. Trigueros, "Qualitative and Quantitative Research tools," *Res. Gate*, no. March 2017, pp. 1–16, 2017, [Online]. Available: https://www.researchgate.net/publication/323014697.

[47] Z. Zhou, Q. Zhi, S. Morisaki, and S. Yamamoto, "An Evaluation of Quantitative Non-Functional Requirements Assurance Using ArchiMate," *IEEE Access*, vol. 8, pp. 72395–72410, 2020, doi: 10.1109/ACCESS.2020.2987964.

[48] A. Rashwan, O. Ormandjieva, and R. Witte, "Ontology-based classification of non-functional requirements in software specifications: A new corpus and SVM-based classifier," *Proc. - Int. Comput. Softw. Appl. Conf.*, no. ii, pp. 381–386, 2013, doi: 10.1109/COMPSAC.2013.64.

[49] K. Lauenroth, E. Kamsties, and O. Hehlert, "Do Words Make a Difference? An Empirical Study on the Impact of Taxonomies on the Classification of Requirements," *Proc. - 2017 IEEE 25th Int. Requir. Eng. Conf. RE 2017*, pp. 273–282, 2017, doi: 10.1109/RE.2017.57.

[50] C. Pan, M. Lu, H. Zhang, and B. Xu, "Qualitative Software Reliability Requirements: Concept, Classification and Practical Elicitation Methods," *Proc. - 2018 IEEE 18th Int.*

Conf. Softw. Qual. Reliab. Secur. Companion, QRS-C 2018, pp. 164–171, 2018, doi: 10.1109/QRS-C.2018.00040.

[51] Y. Wang and J. Zhang, "Experiment on automatic functional requirements analysis with the EFRFs semantic cases," *PIC 2016 - Proc. 2016 IEEE Int. Conf. Prog. Informatics Comput.*, pp. 636–642, 2017, doi: 10.1109/PIC.2016.7949577.

[52] S. Abualhaija, C. Arora, M. Sabetzadeh, L. C. Briand, and M. Traynor, "Automated demarcation of requirements in textual specifications: a machine learning-based approach," *Empir. Softw. Eng.*, vol. 25, no. 6, pp. 5454–5497, 2020, doi: 10.1007/s10664-020-09864-1.

[53] R. Sonbol, G. Rebdawi, and N. Ghneim, "Towards a Semantic Representation for Functional Software Requirements," *Proc. - 7th Int. Work. Artif. Intell. Requir. Eng. AIRE 2020*, pp. 1–8, 2020, doi: 10.1109/AIRE51212.2020.00007.

[54] K. Ayub, F. Azam, M. W. Anwar, A. Amjad, and M. S. Jahan, "A Novel Approach for Software Requirement Prioritization Based Upon Non Functional Requirements," *Proc. - 2019 7th Int. Conf. Softw. Eng. Res. Innov. CONISOFT 2019*, pp. 8–15, 2019, doi: 10.1109/CONISOFT.2019.00013.

[55] N. Niu and S. Easterbrook, "Extracting and modeling product line functional requirements," *Proc. 16th IEEE Int. Requir. Eng. Conf. RE'08*, pp. 155–164, 2008, doi: 10.1109/RE.2008.49.

[56] N. Cherdsakulwong and T. Suwannasart, "Impact Analysis of Test Cases for Changing Inputs or Outputs of Functional Requirements," *Proc. - 20th IEEE/ACIS Int. Conf. Softw. Eng. Artif. Intell. Netw. Parallel/Distributed Comput. SNPD 2019*, pp. 179–183, 2019, doi: 10.1109/SNPD.2019.8935754.

[57] Z. S. H. Abad, A. Shymka, S. Pant, A. Currie, and G. Ruhe, "What are practitioners asking about requirements engineering? An exploratory analysis of social Q&A sites," *Proc. - 2016 IEEE 24th Int. Requir. Eng. Conf. Work. REW 2016*, pp. 334–343, 2017, doi: 10.1109/REW.2016.46.

[58] S. N. Sayyed, A. M. Pundge, and C. Namrata Mahender, "Text mining for proper

identification of noun using Conceptual Dependency," *Proc. Int. Conf. Smart Syst. Inven. Technol. ICSSIT 2018*, no. Icssit, pp. 463–467, 2018, doi: 10.1109/ICSSIT.2018.8748682.

[59] R. B. Johnson and A. J. Onwuegbuzie, "Toward a Definition of Mixed Methods Research," *J. Mix. Methods Res.*, vol. 1, no. 2, pp. 112–133, 2007, doi: 10.1177/1558689806298224.

[60] S. Abrahão, C. Gravino, E. Insfran, G. Scanniello, and G. Tortora, "Assessing the effectiveness of sequence diagrams in the comprehension of functional requirements: Results from a family of five experiments," *IEEE Trans. Softw. Eng.*, vol. 39, no. 3, pp. 327–342, 2013, doi: 10.1109/TSE.2012.27.

[61] Fazal-E-Amin, A. K. Mahmood, and A. Oxley, "A mixed method study to identify factors affecting software reusability in reuse intensive development," *2011 Natl. Postgrad. Conf. - Energy Sustain. Explor. Innov. Minds, NPC 2011*, 2011, doi: 10.1109/NatPC.2011.6136324.

[62] F. Khalique, W. H. Butt, and S. A. Khan, "Creating Domain Non-functional Requirements Software Product Line Engineering Using Model Transformations," *Proc. - 2017 Int. Conf. Front. Inf. Technol. FIT 2017*, vol. 2017-Janua, pp. 41–45, 2017, doi: 10.1109/FIT.2017.00015.

[63] S. L. Buitron, F. J. Pino, and T. R. Curieux, "Using Design-science for the Representation of Non Functional Requirements," *Iber. Conf. Inf. Syst. Technol. Cist.*, vol. 2020-June, no. June, pp. 24–27, 2020, doi: 10.23919/CISTI49556.2020.9140876.

[64] D. García-López, M. Segura-Morales, and E. Loza-Aguirre, "Improving the quality and quantity of functional and non-functional requirements obtained during requirements elicitation stage for the development of e-commerce mobile applications: An alternative reference process model," *IET Softw.*, vol. 14, no. 2, pp. 148–158, 2020, doi: 10.1049/iet-sen.2018.5443.

[65] H. Sharp, Y. Dittrich, and C. R. B. De Souza, "The role of ethnographic studies in empirical software engineering," *IEEE Trans. Softw. Eng.*, vol. 42, no. 8, pp. 786–804,

2016, doi: 10.1109/TSE.2016.2519887.

[66] A. N. Ghazi, K. Petersen, S. S. V. R. Reddy, and H. Nekkanti, “Survey research in software engineering: Problems and mitigation strategies,” *IEEE Access*, vol. 7, no. c, pp. 24703–24718, 2019, doi: 10.1109/ACCESS.2018.2881041.

[67] Y. Wang, “Automatic semantic analysis of software requirements through machine learning and ontology approach,” *J. Shanghai Jiaotong Univ.*, vol. 21, no. 6, pp. 692–701, 2016, doi: 10.1007/s12204-016-1783-3.

[68] A. Ranjan Pal and D. Saha, “Word Sense Disambiguation: A Survey,” *Int. J. Control Theory Comput. Model.*, vol. 5, no. 3, pp. 1–16, 2015, doi: 10.5121/ijctcm.2015.5301.

[69] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology*. Sage, 2004.

[70] N. Choudhary and B. K. Jesiek, “State of mixed methods research in engineering education: In-depth examination of JEE articles, 2010-2015,” *Proc. - Front. Educ. Conf. FIE*, vol. 2016-Novem, pp. 2010–2015, 2016, doi: 10.1109/FIE.2016.7757554.

[71] M. Di Penta and D. A. Tamburri, “Combining quantitative and qualitative studies in empirical software engineering research,” *Proc. - 2017 IEEE/ACM 39th Int. Conf. Softw. Eng. Companion, ICSE-C 2017*, pp. 499–500, 2017, doi: 10.1109/ICSE-C.2017.163.

[72] K. Krippendorff, *Content Analysis: An Introduction to Its Methodology*. TPB, 1996.

[73] J. Maxwell, “Qualitative Research Design : An Interactive Approach / J.A. Maxwell.,” 2012.

[74] R. Hall, “Mixed methods: In search of a paradigm,” *Conduct. Res. a Chang. Challenging World*, no. March, pp. 69–78, 2013.

[75] E. Klotins, “Using the Case Survey Method to Explore Engineering Practices in Software Start-Ups,” *Proc. - 2017 IEEE/ACM 1st Int. Work. Softw. Eng. Startups, SoftStart 2017*, pp. 24–26, 2017, doi: 10.1109/SoftStart.2017.4.

[76] B. Kitchenham and S. Charters, “Guidelines for performing Systematic Literature

Reviews in SE," pp. 1–44, 2007, doi: 10.1145/1134285.1134500.

[77] S. Easterbrook, "Empirical research methods for software engineering," *ASE'07 - 2007 ACM/IEEE Int. Conf. Autom. Softw. Eng.*, p. 574, 2007, doi: 10.1145/1321631.1321749.

[78] B. Weights and P. Data, "StarTips."

[79] M. Tavakol and R. Dennick, "Making sense of Cronbach's alpha," *Int. J. Med. Educ.*, vol. 2, pp. 53–55, 2011, doi: 10.5116/ijme.4dfb.8dfd.

[80] Abdul, "Quality of Psychology Test Between Likert Scale 5 and 6 Points," *J. Soc. Sci.*, vol. 6, no. 3, pp. 399–403, 2010, doi: 10.3844/jssp.2010.399.403.

[81] J. M. Cortina, "What Is Coefficient Alpha? An Examination of Theory and Applications," *J. Appl. Psychol.*, vol. 78, no. 1, pp. 98–104, 1993, doi: 10.1037/0021-9010.78.1.98.

[82] N. Carr, G. W. Gray, and D. G. McDonnell, "1-(TRANS-4 prime -N-ALKYLCYCLOHEXYL)-2-(4 double prime -CYANOPHENYL)ETHANES - A NEW SERIES OF STABLE NEMATOGENS OF POSITIVE DIELECTRIC ANISOTROPY.," *Mol. Cryst. Liq. Cryst.*, vol. 97, no. 1–4, pp. 13–28, 1983, doi: 10.1080/00268948308073138.